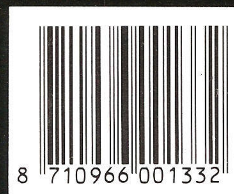
Commodore INFO

PRIJS ƒ 7.95 / Bfr. 160



ONAFHANKELIJK BLAD VOOR COMMODORE GEBRUIKERS Jaargang 7, NO.2, maart/april '90

LISTINGS:
Lingo (64)
Rekenwonder (64)

Amiga voor hobby en professie

Nordic Power:
Cartridge voor de Amiga

PD-Software tegen virussen

Filerequester voor de Amiga

Met Tips en Trucs 64, Oud van Goud, Miniatuurtjes, graphics op de 64 en listings voor de 64

Commodore Info

Verschijnt 8x per jaar
Jaarg.7, no.2, mrt/apr. 1990

Uitgave: Sala Communications
Uitgever: J. Broekhuizen
Redactie:
ir. L. Sala hoofdredacteur
J. Bodzinga adj. hoofdred.
drs. J. Boers eindredacteur
H. Smeenk, drs. U. Schuurmans, W.A. Scheer R. Goudriaan, B. Venema, P. Boncz

Produktie:
drs. H. Zoete, J. Broekhuizen, M. van Zijl

Strip: Bert Tier
Illustraties: Ben van Mierlo

Redactie adres:
Postbus 43048, 1009 ZA Amsterdam
tel. 020-228871

Listingtelefoon: (ma: 17.00-21.00 u)
02155-25162

Advertentie-exploitatie:
Ing. V. Sala, Ing. B. Sala, D. van Vlijmen
Postbus 43048, 1009 ZA Amsterdam
tel. 020-273198

Abonnementen en administratie:
Sandra v.d. Meyden en Marjo Jansen
Postbus 43048
1009 ZA Amsterdam
tel. 020-248006
Vragen betreffende abonnementen ontvangen wij bij voorkeur schriftelijk, met meesturen van het omslagetiket.

Abonnement:
Voor 8 nummers *f* 47,50 of Bfr. 975 per jaar. Betaling op giro 4985259 (België: BBL nr. 310050602562) t.n.v. SAC/Commodore-Info. Oude nummers kunt U alleen krijgen bij vooruitbetaling van *f* 6,75 op de bovenstaande rekening. Ook telefonische opgave voor een abonnement is mogelijk. Bel GRATIS 06-02242222 (teleservice), elke dag tot 20.20 uur (dus ook in het weekend). België: 115555, dagelijks tot 22.00 uur. Deze telefoonnummers zijn alleen bedoeld voor opgave van NIEUWE abonnementen.
Opzegging dient schriftelijk te geschieden uiterlijk twee maanden voor de aanvang van een nieuwe abonnementsperiode van een jaar.

Vragen over artikelen:
Voor vragen en opmerkingen over artikelen of anderszins graag een briefje aan het redactie-adres (zie boven).

Cover: DeLuxe Photolab (Amiga)

Imagesetting IFF: 3Gitaal, Amsterdam

Zetwerk & druk: NDB, Zoeterwoude

Distributie:
In Nederland: Betapress, Gilze
In België: AMP, Brussel

ISSN: 0169-3085

Inhoud van dit nummer

Johan & Johan
Voor dit nummer ontvingen wij geen kopij van Johan & Johan. In nummer 3 hopen wij dit op te nemen.

Redactioneel

We zijn met Commodore Info al weer heel wat jaren bezig. Ik herinner me nog hoe in voorjaar 1983 het eerste nummer uitkwam. Toen was er heel wat discussie over hoe je als blad zowel de Vic-20 gebruikers als degenen, die de toen nieuwe C-64 hadden aangeschaft, tegelijk kon bedienen. En in die dagen was Commodore heel wat agressiever met nieuwe produkten, we hadden toen nog geregeld nieuwe zakelijke 8-bitters, zoals de 8296 en de 9000 familie, om nog maar niet te spreken over de Vic-10, de C-16 en Plus-4 en later de C-128. Ondertussen zijn we, anno 1990, nog steeds in dubio over hoeveel aandacht we moeten besteden aan de Amiga.
Onze thuisbasis is duidelijk de C-64, maar heel wat Amiga gebruikers vinden dat ze ook in Commodore-Info thuis horen. Ook Commodore zelf wil graag dat we wat meer aan de Amiga familie doen, maar aan de andere kant zijn er ook aparte Amiga bladen. De meest professionele daarvan, de Nederlandse Amiga (Wereld)World uitgave, is overigens na een paar nummers al weer gesneuveld. En daar zitten we dan weer. Hoe beïnvloedt dat Commodore Info? We kiezen voor de gulden middenweg, dit nummer heeft een tamelijk zwaar Amiga accent en we zullen zien, hoe de lezers dat waarderen. Mogelijk zetten we die trend voort, dat hangt ook af van het commerciële draagvlak. In ieder geval is er dit voorjaar interessant Amiga nieuws, een nieuw model met CD-ROM gaat deze machine vast extra interessant maken.

Luc Sala

Graphics op de 64

Deel 8: animatie

Na alle theoretische verhandelingen over de grafische mogelijkheden op de Commodore 64, zijn Michel de Boer en Hylke Sprangers aangekomen bij het meest fascinerende facet van graphics.... ANIMATIE. In deze aflevering zullen de sprites rondhollen, de karakters rondtollen en de schermen smooth scrollen. Kortom, deze aflevering bevat alle basis ingredienten om flitsende demo's en spetterende video games te bouwen.

Het menselijk oog heeft een beperkte verwerkingssnelheid. Door snel een serie beelden, die een klein beetje van elkaar verschillen, achter elkaar te projecteren, neemt het oog niet elk beeldje als een afzonderlijk beeld waar, maar ontstaat de illusie van beweging. In deze aflevering zullen we uitleggen hoe met de verschillende grafische objecten deze illusie verkregen kan worden. Daarbij zullen de volgende onderwerpen de revue passeren. Het eerste onderwerp is smooth scrolling. Daarna zullen we het hebben over het bewegen van sprites en animatie van karakters.

Scrolling

In de vorige aflevering hebben we een begin gemaakt met het onderwerp scrolling. Dit is een vorm van animatie waarbij het hele scherm (of een aantal regels) in horizontale of vertikale richting wordt verschoven. We hebben toen twee soorten scrolling onderscheiden: coarse scrolling en smooth scrolling.
Bij coarse scrolling wordt het scherm steeds een heel karakter opgeschoven. De animatie die hierdoor ontstaat is vrij grof. Coarse scrolling hebben we in de vorige aflevering uitgebreid behandeld.



Bij de tweede soort scrolling, smooth scrolling, wordt het scherm steeds een pixel in plaats van een heel karakter verschoven. Hierdoor kan een mooie vloeiende beweging van het scherm gecreeerd worden.

We zullen nog even kort samenvatten wat we de vorige keer over smooth scrolling hebben verteld. In figuur 1 staan alle adressen die nodig zijn voor smooth scrolling. Bit 7 van adres 53265 en adres 53266 worden nog besproken. Voor zowel vertikale als horizontale scrolling zijn drie bits beschikbaar. Als in deze drie bits van 0 tot 7 wordt geteld, wordt afhankelijk van het gebruikte adres omlaag of naar rechts gescrolld. Als van 7 tot 0 wordt teruggeteld, wordt er omhoog of naar links gescrolld. Nadat het scherm op die manier 7 pixels verschoven is, moet met coarse scrolling het scherm een karakter opgeschoven worden en weer opnieuw met het tellen begonnen worden. Om er voor te zorgen dat de zijkant van het scherm niet trilt tijdens het scrollen, moet de border verbreed worden.
Als laatste voorbeeld hebben we een programma laten zien dat een heel scherm, gevuld met gelijke karakters, smooth

Adressen voor smooth scrolling	
Adres	Functie
53270	
bit 0-2	horizontaal scrollen
bit 3	1 = normale zijborder 0 = brede zijborder
53265	
bit 0-2	vertikaal scrollen
bit 3	1 = normale boven- en onderborder 0 = brede boven- en onderborder
bit 7	extra bit voor rasterlijn nummer
53266	rasterlijn nummer

Figuur 1.

scrollt. In dat programma zat nog een onvolkomenheid. Het scrollen verliep niet helemaal soepel, maar golfde een beetje. Om het golven te verhelpen moeten we weten hoe het beeldscherm wordt opgebouwd.

Rasterlijnen

Het beeldscherm is verdeeld in 312 horizontale lijnen. Deze lijnen worden rasterlijnen genoemd. Een rasterlijn heeft de dikte van een pixel. De rasterlijnen zijn genummerd van 0 tot 311. Lijn 0 bevindt zich boven aan het scherm. Het scherm wordt 50 keer per seconde van boven naar beneden opgebouwd. Elke rasterlijn wordt daarbij van links naar rechts opgebouwd. Rasterlijnen 50-249 komen overeen met het gedeelte binnen de borders. Als de opbouw nu net bij een van deze lijnen (bijvoorbeeld lijn 100) is en we scrollen het scherm, dan zal het scherm vanaf lijn 100 direct verschoven worden, terwijl lijnen 50-99 pas in de volgende opbouw cyclus worden verschoven. Hierdoor ontstaat de golving.
Om de golving te voorkomen, moeten we er dus voor zorgen dat we het scherm pas scrollen, als de opbouw voorbij lijn 249 is, zodat de verschuiving van het scherm pas in de volgende opbouw cyclus zichtbaar wordt. In adres 53266 wordt bijgehouden bij welke rasterlijn de opbouw is. In dit adres kunnen alleen getallen tussen 0 en 255 staan. Om rasterlijnen groter dan 255 aan te geven wordt bit 7 van adres 53265 gebruikt. Als dit bit uit staat, is de waarde in adres 53266 het nummer van de rasterlijn. Als dit bit aan staat, dan moet er 256 bij de waarde van adres 53266 worden opgeteld. Als er in 53266 bijvoorbeeld 10 staat en bit 7 van adres 53265 staat aan, dan is de opbouw bij rasterlijn 266. Door deze adressen te testen, kunnen we bij de juiste rasterlijn beginnen met scrollen. We kunnen bijvoorbeeld steeds de waarde van adres 53266 uitlezen en kijken of deze waarde gelijk aan 249 is. Als dit niet het geval is, herhalen we de handeling, anders scrollen we het scherm. We hoeven in dit geval geen rekening te houden met bit 7 in adres 53265, omdat deze nooit aan kan staan als adres 53266 de waarde 249 bevat.
Alle voorbeeldprogramma's uit de vorige aflevering hadden steeds een beperking, omdat ze in Basic waren geschreven; ze scrollden maar een regel of een heel scherm met dezelfde karakters. We geven daarom een assembly programma dat een heel scherm met verschillende karakters naar links scrollt. Op elke regel van het scherm scrollt een lichtkrant.
In machinetaal is het ook niet zo makkelijk om de juiste snelheid te bereiken. Er moet namelijk voor gezorgd worden dat de coarse scrolling minstens net zo snel is als een scherm opbouw cylus; dus binnen 1/50 seconde. Als dit niet het geval is dan haalt het opbouwen het scrollen in en wordt het onderste gedeelte van het scherm al afgebeeld voordat het naar links geschoven is. Om de listing niet al te lang en onoverzichtelijk te maken, hebben we voor de coarse scrolling een andere methode gebruikt, dan we beschreven hebben. Deze methode schuift niet steeds alle karakters een plaats naar links, maar werkt als volgt. Eerst worden de karakters 0 tot 39 van de lichtkrant op het scherm gezet. Vervolgens de karakters 1-40, dan 2-41 enz. Als we aan het einde van de lichtkrant zijn, dan beginnen we weer met karakters 0-40. Door er nu voor te zorgen dat de laatste 39 karakters gelijk zijn aan de eerste 39, ontstaat hetzelfde resultaat als bij de beschreven methode. We schui-

```
;*******************
;* SMOOTH SCOLLING *
;*******************
;
;*** initialisatie
        sei
        lda #7
        sta scroll
;*** $fd/fe bevat het
    beginadres van
;*** het window dat over de
    tekst
;*** wordt geschoven
        lda #<text
        sta $fd
        lda #>text
        sta $fe
;*** zet scherm op 38 kolommen
        lda 53270
        and #240
        ora #7
        sta 53270
;*** wacht op juiste rasterlijn
wait    lda 53266
        cmp #250
        bne wait
;*** scroll scherm
        lda scroll
        beq w2
        dec scroll
        lda 53270
        and #248
        ora scroll
        sta 53270
;*** wacht tot opbouw voorbij
;*** rasterlijn 250 is
w1      lda 53266
        cmp #250
        beq w1
        jmp wait
;*** schuif scherm terug
;*** naar rechts
w2      lda #7
        sta scroll
        ora 53270
        sta 53270
;*** coarse scrolling naar
    links
        jsr coarse
;*** schuif window naar rechts
        inc $fd
        bne test
        inc $fe
;*** test of einde van
;*** lichtkrant bereikt is
test    ldy #39
        lda ($fd),y
        cmp #$ff
        bne wait
        lda #<text
        sta $fd
        lda #>text
        sta $fe
        jmp wait
;*** schuif scherm een karakter
;*** naar links
coarse  lda #<1024
        sta c3+1
        lda #>1024
        sta c3+2
        ldx #25
        ldy #0
c1      lda ($fd),y
c3      sta $0400,y
        iny
        cpy #40
        bne c1
        ldy #0
        clc
        lda c3+1
        adc #40
        sta c3+1
        bcc c2
        inc c3+2
c2      dex
        bne c1
        rts
;*** teller voor bits 0-2
;*** van adres 53270
scroll .byt 0
;*** schermcodes
text .byt
    32,32,32,32,32,32,32,32
    .byt 32,32,32,32,32,32,32,32
    .byt 31,32,32,32,32,32,32,32
    .byt 32,32,32,32,32,32,32,32
    .byt 32,32,32,32,32,32,32,32
    .byt 19,13,15,15,20,8,32
    .byt 19,3,18,15,12,12,9,14,7
    .byt 32,32,32,32,32,32,32,32
    .byt 32,32,32,32,32,32,32,32
    .byt 31,32,32,32,32,32,32,32
    .byt 32,32,32,32,32,32,32,32
    .byt 32,32,32,32,32,32,32
    .byt 255
```

Listing Smooth scrolling



Figuur 2.

ven als het ware een window van 40 karakters breed naar rechts over de tekst. Alle karakters binnen dit window zijn zichtbaar op het scherm. Figuur 2 illustreert de methode.
U kunt deze listing intikken met behulp van een assembler; bijvoorbeeld met PROFI of PAL assembler.

Sprite animatie

Sprites hebben op het gebied van animatie het meest te bieden. In tegenstelling tot een karakter, kan een sprite op elke willekeurige positie op het scherm worden geplaatst. Om de sprite vervolgens op een nieuwe positie neer te zetten, hoeft deze niet eerst uitgewist te worden op zijn oude positie. Het enige dat gedaan hoeft te worden, is het doorgeven van de nieuwe positie aan de video chip. De video chip zorgt er dan voor dat de sprite van de oude positie verdwijnt en terecht komt op de nieuwe positie, zonder dat daarbij andere grafische objecten, die op het scherm staan, aangetast worden. De sprites overlappen die objecten, of worden door die objecten overlapt. Hierdoor zijn sprites uitermate geschikt voor het bewegen van figuurtjes over een achtergrond van karakters.

Bewegen van sprites

Het bewegen van sprites is een eenvoudige zaak. Het enige dat gedaan hoeft te worden is het veranderen van de x- en y-coördinaat van de sprite. Het volgende programma laat sprite 0 in een cirkel over het scherm bewegen. Om de listing niet onnodig lang te maken met bytewaarden voor de vorm van de sprite, hebben we als vorm gewoon een rechthoek genomen.

```
1 rem sprite beweging
5 dim x(300),y(300)
10 for i=832 to 894:poke i,255:next
20 poke 53280,0:poke 53281,0
30 for i=0 to 2*{pi} step {pi}/60
40 x(a)=50*cos(i)+124
50 y(a)=50*sin(i)+150
60 a=a+1
70 next
80 poke 2040,13:poke 53269,1
90 for i=0 to a-1
100 poke 53248,x(i):poke 53249,y(i)
110 next
120 goto 90
```

Regels 30-70 berekenen alle coördinaten van de cirkelbaan waarlangs de sprite zich beweegt. Deze coördinaten worden opgeslagen in twee arrays. Regels 90-120 poken deze coördinaten achtereenvolgens in de coördinaat registers voor sprite 0.
In plaats van eerst alle coördinaten berekenen en opslaan, hadden we de coördinaten ook direct in de coördinaat registers kunnen poken. Hierdoor zou een hoop geheugenruimte bespaart worden en hadden we niet zo lang hoeven te wachten tot de beweging begon. Een heel groot nadeel van die methode is dat de beweging zeer traag wordt door de ingewikkelde berekening. Daarom hebben we ervoor gekozen om eerst alle coördinaten te berekenen. Als de berekening heel eenvoudig is, dan werkt de andere methode meestal wel snel genoeg.
Door de afstand tussen twee opvolgende coördinaten te vergroten, kan een hogere snelheid gehaald worden. In het voorbeeld programma moet u dan de stapgrootte in regel 30 vergroten. Dit gaat echter wel ten koste van de soepelheid van de beweging.
In het voorbeeld programma hebben we de x-coördinaat steeds in adres 53248 gepoked. Dit gaat goed, omdat deze coördinaat hier altijd kleiner dan 255 is. Als de x-coördinaat echter groter dan 255 kan zijn, moet u ervoor zorgen dat het MSB bit in 53264 de juiste waarde heeft.

Animeren van sprites

Om de beweging van een sprite een beetje realistisch te maken, is het vaak nodig om, naast het veranderen van de positie , ook de vorm van de sprite te veranderen. Een mannetje dat voor zijn leven over het scherm rent, ziet er immers niet zo realistisch uit als hij daarbij zijn benen niet beweegt. Om het rennen realistisch te maken, moeten we de stand van de benen van het mannetje steeds een beetje veranderen. We zouden dit kunnen doen door steeds nieuwe bytewaarden in de sprite te poken. Dit vreet echter nogal wat tijd. De spritepointers maken het mogelijk om het animeren van een sprite veel sneller te maken. We moeten daarvoor eerst alle standen van de sprite in het geheugen opslaan. Het effect van animatie kan nu verkregen worden door de sprite pointer steeds naar de volgende stand van de sprite te laten wijzen.
Het volgende programma illustreert de methode. Het laat een wiel over het scherm rollen.

```
5 rem sprite animatie
10 for i=832 to 1023
20 read a:poke i,a:next
30 poke 53280,0:poke 53281,0
40 print"{SHIFT-CLR}"
50 poke 53269,1:poke 2040,13
60 poke 53249,150:x=0
70 for i=13 to 15
80 poke 2040,i
90 for j=0 to 1
100 x=x+1:poke 53248,x and 255
110 poke 53264,(x and 256)/256
115 if x=350 then x=0
120 next j,i
130 goto 70
500 rem stand 1
510 data 0,124,0,1,255,0,7,147
520 data 192,14,16,224,28,16,112,26
530 data 16,176,49,17,24,48,146,24
540 data 96,84,12,96,56,12,127,255
550 data 252,96,56,12,96,84,12,48
560 data 146,24,49,17,24,26,16,176
570 data 28,16,112,14,16,224,7,147
580 data 192,1,255,0,0,124,0,0
590 rem stand 2
600 data 0,124,0,1,255,0,7,131
610 data 192,15,4,224,28,132,112,24
620 data 132,48,48,72,56,48,72,216
630 data 110,51,12,97,148,12,96,124
640 data 12,96,83,12,97,152,236,54
650 data 36,24,56,36,24,24,66,48
660 data 28,66,112,14,65,224,7,131
670 data 192,1,255,0,0,124,0,0
680 rem stand 3
690 data 0,124,0,1,255,0,7,131
700 data 192,14,65,224,28,66,112,24
710 data 66,48,56,36,24,54,40,24
720 data 97,40,236,96,215,12,96,56
730 data 12,97,214,12,110,41,12,48
740 data 40,216,48,72,56,24,132,48
750 data 28,132,112,15,4,224,7,131
760 data 192,1,255,0,0,124,0,0
```

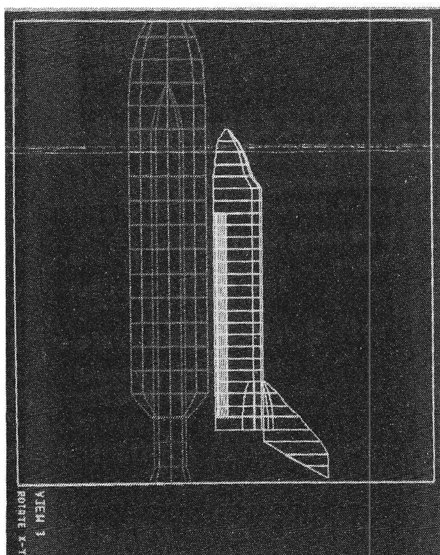
In regel 80 wordt de spritepointer steeds op de volgende stand van het wiel gezet. Regels 100-110 bewegen de sprite naar rechts over het scherm. De berekening voor deze beweging is zo simpel dat deze real time kan worden uitgevoerd. We hoeven niet een heel array met waarden te vullen, zoals bij het vorige programma. De rest van het programma spreekt voor zich.

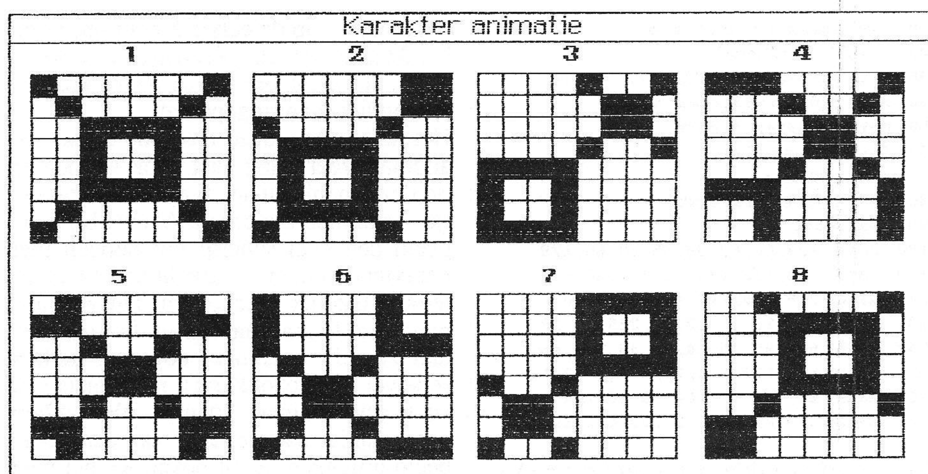
Nadeel van Basic

Basic is eigenlijk niet zo geschikt voor het programmeren van animatie. Het is erg traag en als u de voorbeeld programma's uitgeprobeerd hebt, zult u wel gemerkt hebben dat de sprites af en toe knipperen of golven. Om dit te voorkomen moet er net als bij scrolling rekening worden gehouden met de schermopbouw. Als u ervoor zorgt dat u pas iets op het scherm verandert, als de rasterlijn buiten het scherm is, dan zal de animatie soepel verlopen. Dit kan uiteraard alleen vanuit machinetaal bewerkstelligd worden. De reden dat we de programma's toch in Basic hebben geschreven is, dat Basic de animatie technieken beter illustreert. Met enige kennis van machinetaal moet u in staat zijn om deze programma's in machinetaal te herschrijven.

Tip voor 3D animatie

Door handig gebruik te maken van overlappingsprioriteiten, kunt u mooie 3D animaties maken. Stel u heeft een sprite van een hond en een sprite van boom en u wilt de hond om de boom laten hollen. De hond moet dus eerst voor de boom en daarna achter de boom langs hollen. U zet dan eerst de hond in sprite 0 en de boom in sprite 1. De hond kan nu voor de boom langs hollen, omdat sprite 0 een hogere prioriteit heeft dan sprite 1. Daarna zet u de hond in sprite 1 en de boom in sprite 0 door de waarde van de spritepointers van beide sprites te verwisselen. Daarbij moet u ook de inhoud van alle andere adressen die bij de sprites horen verwisselen, zoals de coördinaten en de kleur. Als u dit allemaal doet terwijl de rasterlijn buiten het scherm is, zal er op het scherm geen verandering te zien zijn. Maar omdat de hond nu sprite 1 is en de boom sprite 0, kan de hond nu achter de boom langs hollen en is het 3D effect bereikt.





Figuur 3

Door multicolor karakters en sprites te gebruiken, kunnen ook mooie 3D effecten gemaakt worden. Bij multicolor karakters wordt de kleur met bitcombinatie 01 namelijk altijd door sprites overlapt. De andere kleuren worden afhankelijk van de overlappingsprioriteit overlapt. Als we sprites nu een lagere prioriteit dan karakters geven, dan kan een sprite voor een figuur met kleur 01 bewegen en achter een figuur in de andere kleuren. Als u bijvoorbeeld een boom in kleur 01 maakt en voor deze boom een hekje in de andere kleuren plaatst, dan kan de hond uit het voorgaande voorbeeld tussen de boom en het hekje lopen.

Animatie van karakters

In de vorige aflevering hebben we het over coarse scrolling gehad. Dit is een manier van animatie van karakters. Het scherm wordt dan verschoven over de breedte of lengte van een karakter. Het verschuiven van een scherm met karakters wordt verkregen door alle karakters afzonderlijk op te schuiven. We kunnen bijvoorbeeld de karakters op de bovenste regel van het scherm telkens een plaatsje naar links opschuiven. Het effect van een lichtkrant wordt dan verkregen, hetzij wel zeer schokkerig.

Door gewoon karakters over het scherm te laten bewegen, kan eenvoudige animatie van karakters worden verkregen. We geven een voorbeeld. We zetten ergens op het scherm een balletje neer. Door dit balletje steeds eerst uit te wissen (een spatie over het balletje heen poken), en vervolgens op een aangrenzende schermplaats neer te zetten, onstaat simpele animatie. Van deze simpele animatie geven we geen voorbeeld meer, zie hiervoor eventueel de vorige aflevering van deze cursus.

Het nadeel van de hiervoor beschreven animatietechniek voor karakters, is dat het hele groffe animatie is. Het bewegen van een figuur, tekst of iets dergelijks gebeurt immers steeds met acht pixels tegelijk. Wat we dus willen is animatie die vloeiend verloopt. De figuren op het scherm moeten niet per 8, maar per 1 pixel tegelijk verschoven worden. In dat geval komen we al snel uit bij smooth scrolling, want bij smooth scrolling wordt het scherm slechts een pixel per keer verschoven. We kunnen echter een soortgelijk effect ook op andere manieren verkrijgen.

De eerste manier is om eerst een aantal karakters van te voren te ontwerpen. Door nu de verschillende karakters telkens over elkaar heen te poken, kan animatie worden verkregen. we geven een voorbeeld, om dit wat duidelijker te maken. In figuur 3 staan acht karakters getekend. Het eerste karakter is een vierkant met op de vier hoeken schuine streepjes. Het tweede karakters laat hetzelfde motief zien als het eerste karakter, met het verschil dat het figuurtje een pixel naar linksonder is verschoven. In het derde karakter is het figuurtje weer een stapje naar linksonder verschoven. Dit gaat zo door tot het achtste karakter. Als het achtste karakter weer een pixel naar linksonder wordt verschoven, wordt weer het eerste karakter verkregen.

Als acht karakters worden herdefinieerd tot deze acht zelfgemaakte karakters, kunnen we vervolgens betrekkelijk simpel animatie verkrijgen. Want we kunnen nu deze acht karakters achterelkaar op een bepaalde schermplaats zetten. Het volgende programma laat hier een voorbeeld van zien:

```
10 rem karakter animatie
20 print"{clr/home}"
30 poke 53272, peek(53272) or 14
40 for x=0 to 63:read a
50 poke 14336+x,a:next
60 for x=0 to 7
70 poke 1024,x
80 for y=0 to 10:next
90 next:goto 60
100 data 129,66,60,36,36,60,66,129
110 data 3,3,132,120,72,72,120,132
120 data 9,6,6,9,240,144,144,240
130 data 225,18,12,12,18,225,33,33
140 data 66,195,36,24,24,36,195,66
150 data 132,132,135,72,48,48,72,135
160 data 15,9,9,15,144,96,96,144
170 data 33,30,18,18,30,33,192,192
```

In regel 30 wordt gedefinieerd dat de karakterset op adres 14336 begint. In de regels 40 en 50 worden de acht zelfgemaakte karakters op de eerste 8 plaatsen in de karakterset gezet. Vervolgens worden deze acht karakters om de beurt linksboven op het scherm gepoked.
Het nadeel van deze methode is dat het allemaal zeer traag gaat. Ook kunnen er op deze manier maar zeer kleine figuurtjes geanimeerd worden. Als we bijvoorbeeld een heel scherm vol hebben met karakters, kunnen we die in basic niet allemaal tegelijk veranderen. Dit zou immers veel te langzaam gaan. In machinetaal zou dit eventueel nog wel lukken (hoewel wel schokkerig), maar er is een veel eenvoudigere manier. We zullen deze manier aan de hand van een voorbeeld uitleggen. Eerst poken we het hele scherm vol met schermcode 0. Als we niets aan de karakterset hadden gedaan, zou het hele scherm nu vol staan met at-signs. We definieren nu dat de karakterset begint op adres 14336, net zoals in het vorige programma. We kunnen de karakterset ook wel ergens anders in het geheugen laten beginnen, maar dat is nu niet ter zake doend. Zoals al gezegd werken we met het at-sign, die schermcode 0 heeft. Dit karakter beslaat dus de eerste acht bytes van de karakterset, beginnend vanaf adres 14336. Nu gaan we echter het schermgeheugen niet steeds veranderen (dit deden we met adres 1024 in het vorige programma), maar we gaan telkens het karaktergeheugen veranderen. We poken de acht karakters die in figuur 3 staan, steeds om de beurt in karakter 0, het at-sign. We moeten dus steeds acht pokes uitvoeren, dan een tijdje wachten, en dan

Rectificatie graphics

In de vorige twee nummers (NO. 8, dec '89/jan '90 en NO. 1, feb/mrt '90) zijn er helaas een paar zetfouten in de artikelen "Graphics op de 64" en "Tips & Trucs 64" geslopen. Deze fouten betroffen het wegvallen van de kleiner en groter dan tekens. Tevens waren de illustraties behorende bij "Graphics op de 64" niet afgedrukt. De redactie biedt hierbij haar excuses aan aan zowel lezers als schrijvers. We hopen het in dit nummer wel goed te doen.

In deel 6 van "Graphics op de 64" (Computer Kunst) waren een aantal kleiner en groter dan tekens verdwenen. In de een na laatste alinea van de paragraaf "lissajous figuren" en in de alinea onder de listing "sinuskrommen" stonden een aantal regels die de gebruiker in acht moest nemen bij het invoeren van variabelen. In deze regels zijn alle kleiner dan tekens weggevallen. Verder werkte de listing "draaiende lijnen" niet goed. Daarom plaatsen we hieronder de listing nog een keer:

```
5 rem draaiende lijnen
10 input"geef lengte lijnstuk";le
20 input"geef draaihoek";dh
30 input"geef toename lijnstuk";tl
70 ho=90:hires 0,1
100 t=ho*{pi}/180
110 x=int(le*sin(t)+160)
120 y=int(le*cos(t)+100)
130 if x<0 or x>319 or y<0 or y>199 then 190
140 if ho=90 then x1=x:y1=y
150 line x1,y1,x,y,1
160 x1=x:y1=y
170 ho=ho+dh:le=le+tl
180 goto 100
190 goto 190
```

In beide afleveringen zijn de machtverheffingstekens veranderd in tildes (slangetjes). Dit moeten dus vertikale pijlen zijn.

Bank Switching

In het vorige artikel in de serie "Graphics op de 64" hebben we het gehad over bank switching. Per abuis zijn toen de illustraties, behorende bij dit artikel, niet geplaatst. Deze drie illustraties plaatsen we derhalve deze aflevering. Omdat bank switching een pittig onderwerp is, zullen we hieronder nog een samenvatting geven.

Video banks

De grafische chip van de Commodore, de VIC, heeft 14 adreslijnen, en kan dus maar 16 K geheugen tegelijk zien (adresseren). Daarom zijn de 65536 bytes die de Commodore heeft, opgesplitst in 4 stukken van elk 16 K. Deze 4 geheugensecties worden ook wel video banks genoemd. De VIC kan slechts uit een bank tegelijk gegevens halen. In figuur 1 staat de geheugenopdeling in 4 banks. Standaard staat de computer in bank 0. Dit betekent dat alle grafische objecten waarmee u werkt, zich in bank 0 bevinden.

Geheugenverdeling in banks

Bank	Bits 0-1 56576	Gebied	A
0	11	0-16383	3
1	10	16384-32767	2
2	01	32768-49151	1
3	00	49152-65535	0

```
Standaard : bank 0
Bank kiezen:
     POKE 56578, PEEK(56578) OR 3
     POKE 56576, PEEK(56576)
                 AND 252 OR A
```

Figuur 1.

Wisselen van bank

Het wisselen van de ene bank naar de andere wordt bank switching genoemd. Dit kan gebeuren door de volgende twee pokes:

de volgende acht bytes in karakter 0 poken. Het resultaat zal zijn dat alle karakters op het scherm constant veranderen. Dit veroorzaakt een scrollende achtergrond. Hier volgt het programma in assembly, dat het bovenstaande voorbeeld implementeert.

```
Labels Mnemonics Commentaar
        lda #$00 ;lobyte scherm.
        sta $fb
        lda #$04 ;hibyte scherm.
        sta $fc
lus1    ldy #$00 ;zet scherm vol
        lda #$00 ;met nullen.
lus2    sta ($fb),y
        iny
        cpy #$00
        bne lus2
        inc $fc
        lda $fc
        cmp #$08
        bne lus1
        lda #$1f ;zet karakterset
        sta $d018;op adres 14336.
lus3    lda #<code
        sta $fb
        lda #>code
        sta $fc
lus4    ldy #$00
lus5    lda ($fb),y ;zet 8 bytes
        sta $3800,y ;in karakter
    0.
        iny
        cpy #$08
        bne lus5
        ldx #$00
lus6    jsr $eeb3 ;vertraag 1 ms.
        inx
        cpx #$18
        bne lus6
lus7    lda $d012 ;houd rekening met
        cmp #$ff   ;rasteropbouw.
        bne lus7
        clc            ;volgende
    karakter.
        lda $fb
        adc #$08
        sta $fb
        lda $fc
        adc #$00
        sta $fc
        lda $fb
        cmp #<code+64 ;laatste
        beq lus3 ;karakter gehad?
        jmp lus4 ;weer opnieuw.
code
.byt $81,$42,$3c,$24,$24,$3c,$42,$81
.byt $03,$03,$84,$78,$48,$48,$78,$84
.byt $09,$06.$06,$09,$f0,$90,$90,$f0
```

```
POKE 56578, PEEK(56578) OR 3
POKE 56576, PEEK(56576) AND  252 OR A
```

Voor de variabele A moet u de waarde invullen die in figuur 1 staat vermeld. Voor bank 0 heeft A bijvoorbeeld de waarde 3. Als een bank wordt uitgekozen, betekent dit, dat alle grafische objecten (scherm, sprites, karakters etc.) in deze bank staan of komen te staan.

Het karakter geheugen

De Commodore 64 kent 256 verschillende karakters. Omdat elk karakter uit 8 bytes is opgebouwd, zijn er 8*256 = 2048 bytes (2 K) nodig om een karakterset in op te slaan. In een bank passen er zo 16/2 = 8 verschillende karaktersets. In figuur 2 staan de 8 gebieden voor bank 0. Voor de andere drie banks moet u steeds het beginadres van de bank optellen bij de gebieden die in figuur 2 staan. Om een gebied uit te kiezen heeft u de volgende poke nodig:

```
POKE 53272, PEEK(53272) AND 240 OR B.
```

De verschillende waarden voor de variabele B staan in figuur 2. De twee standaard karaktersets (lower- en upper-case) beginnen respectievelijk vanaf adres 53248 en adres 55296. Om deze te bereiken moet eerst het videogeheugen uitgeschakeld worden met:

```
POKE 56334, PEEK(56334) AND 254
POKE 1, PEEK(1) AND 251
```

Het videogeheugen kan weer worden aangeschakeld door:

```
POKE 1, PEEK(1) OR 4
POKE 56334,PEEK(56334) OR 1
```

Zoals gezegd staat de Commodore standaard in bank 0. Toch beginnen de standaard karaktersets vanaf adres 53248. De VIC denkt echter dat ze vanaf adres 4096 staan. Daarom is het gebied van adres 4096 tot 8191 karakterROM. Dit betekent dat u in dit gebied geen grafische objecten kan opslaan. Het gebied van adres 36864 tot 40959 in bank 2 beschouwt de VIC ook als ROM. In de banks 1 en 2 bevindt zich geen karakterROM. Dit betekent, dat u in deze twee banks geen beschikking heeft over een standaard karakterset: u moet er zelf een definieren.

Karakterset gebieden in bank 0

Nummer	Bits 1-3 53272	Gebied	B
0	000	0-2047	0
1	001	2048-4095	2
2	010	4096-6143	4
3	011	6144-8191	6
4	100	8192-10239	8
5	101	10240-12287	10
6	110	12288-14335	12
7	111	14336-16383	14

Standaard : gebied 2
Gebied kiezen:
POKE 53272, PEEK(53272) AND 240 OR B
KarakterROM: 4096-8191 en 36864-40959

Figuur 2.

Het schermgeheugen

U kunt zelf bepalen waar het schermgeheugen zich bevindt in de bank. Standaard staat het scherm van adres 1024 tot 2023 in bank 0. Het scherm kan op 16 verschillende plaatsen in een bank worden neergezet. In figuur 3 staan de 16 plaatsen voor bank 0. In de overige drie banks geldt een soortgelijke opdeling in 16 gebieden. U kunt het scherm verplaatsen met de volgende twee pokes:

```
POKE 53272, PEEK(53272) AND 15 OR C
POKE 648, (SA)/256
```

De variabele C kunt u opzoeken in figuur 3. De variabele SA staat voor schermadres. Standaard staat het scherm op adres 1024 tot 2023. Het schermadres (SA) is dan 1024.

Voorbeeld

U wilt bijvoorbeeld werken in bank 2. U voert dan de volgende twee regels in:

```
.byt $e1,$12,$0c,$0c,$12,$e1,$21,$21
.byt $42,$c3,$24,$18,$18,$24,$c3,$42
.byt $84,$84,$87,$48,$30,$30,$48,$87
.byt $0f,$09,$09,$0f,$90,$60,$60,$90
.byt $21,$1e,$12,$12,$1e,$21,$c0,$c0
```

We hebben in het bovenstaande programma rekening gehouden met de rasteropbouw, om de animatie soepel te laten verlopen. In een videogame of intro laat men een dergelijke routine meestal in de interrupt lopen, zodat er een scrollende achtergrond onstaat.
Tot slot geven we bovenstaand assembly programma in Basic, voor degene die niet over een assembler beschikken.

```
5 rem scrollende achtergrond
10 for i= 49152  to 49298
20 read a:poke i,a:s=s+a:next
30 if s=17842 then sys 49152
40 print "fout in data"
1000 data 169,0,133,251,169,4,133,252
1010 data 160,0,169,0,145,251,200,192
1020 data
     0,208,249,230,252,165,252,201
1030 data 8,208,237,169,31,141,24,208
1040 data
     169,83,133,251,169,192,133,252
1050 data 160,0,177,251,153,0,56,200
1060 data 192,8,208,246,162,0,32,179
1070 data
     238,232,224,22,208,248,173,18
1080 data
     208,201,255,208,249,24,165,251
1090 data
     105,8,133,251,201,147,240,208
1100 data 76,40,192,129,66,60,36,36
1110 data 60,66,129,3,3,132,120,72
1120 data 72,120,132,9,6,6,9,240
1130 data 144,144,240,225,18,12,12,18
1140 data 225,33,33,66,195,36,24,24
1150 data 36,195,66,132,132,135,72,48
1160 data 48,72,135,15,9,9,15,144
1170 data 96,96,144,33,30,18,18,30
1180 data 33,192,192
```

Wij hopen dat u met de in deze aflevering behandelde onderwerpen een indruk hebt gekregen van de mogelijkheden van computer animatie op de Commodore 64. Veel plezier bij het maken van uw eigen animatie demo's.

Michel de Boer & Hylke Sprangers

```
POKE 56578, PEEK(56578) OR 3
POKE 56576, PEEK(56576) AND 252 OR 1
```

Nu willen we het schermgeheugen in het derde gebied van bank 2 zetten. Het schermadres is dan 2048 + 32768 = 34816. Dit geven we vervolgens door aan de computer:

Schermgeheugen gebieden in bank 0			
Nummer	Bits 4-7 53272	Gebied	C
0	0000	0-999	0
1	0001	1024-1023	16
2	0010	2048-3047	32
3	0011	3072-4071	48
4	0100	4096-5095	64
5	0101	5120-6119	80
6	0110	6144-7143	96
7	0111	7168-8167	112
8	1000	8192-9191	128
9	1001	9216-10215	144
10	1010	10240-11239	160
11	1011	11264-12263	176
12	1100	12288-13287	192
13	1101	13312-14311	208
14	1110	14336-15335	224
15	1111	15360-16359	240

```
Standaard : gebied 1
Gebied kiezen:
     POKE 53272, PEEK(53272)
                 AND 15 OR C
     POKE 648, (adres scherm)/256
```

Figuur 3.

```
POKE 53272, PEEK(53272) AND 15 OR 32
POKE 648, 34816/256
```

We willen nog even benadrukken dat de VIC de gebieden 4096-8191 (bank 0) en 36864-40959 (bank 2) als karakter-ROM ziet. Op deze gebieden kunnen dus geen grafische objecten staan. Omdat in bank 1 en in bank 3 geen karakterROM is, kunnen in deze twee banks de objecten overal staan. In deze banks heeft u echter geen standaard karakterset tot uw beschikking.

Het kleurengeheugen

Het kleurenhegeheugen kan niet verplaatst worden en loopt altijd van adres 55296 tot adres 56295.

De sprites

Een sprite bestaat uit 63 bytes. Het eerste adres van het geheugengebied waar u de sprites zet, moet een veelvoud zijn van 64. Zo passen er in een bank 16384/64 = 256 sprites. Het gebied waar een sprite staat, kunt u aan de computer doorgeven d.m.v. sprite pointers. De sprite pointers voor de 8 sprites die de Commodore 64 telt, bevinden zich vanaf adres 2040 tot 2047. In deze adressen moet het gebied (0 - 255) gepoked worden, waar de sprite staat. U moet echter oppassen als u het scherm verplaatst, want de sprite pointers verhuizen steeds met het scherm mee. U moet onthouden dat de sprite pointers steeds 17 plaatsen na het laatste schermadres staan. Ook moet u aan het volgende denken: Standaard staat het scherm in gebied 1 van bank 0. Als u nu bijvoorbeeld naar bank 2 overschakelt, staat het scherm nog steeds in gebied 1, echter nu in gebied 1 van bank 2!

Het high resolution scherm

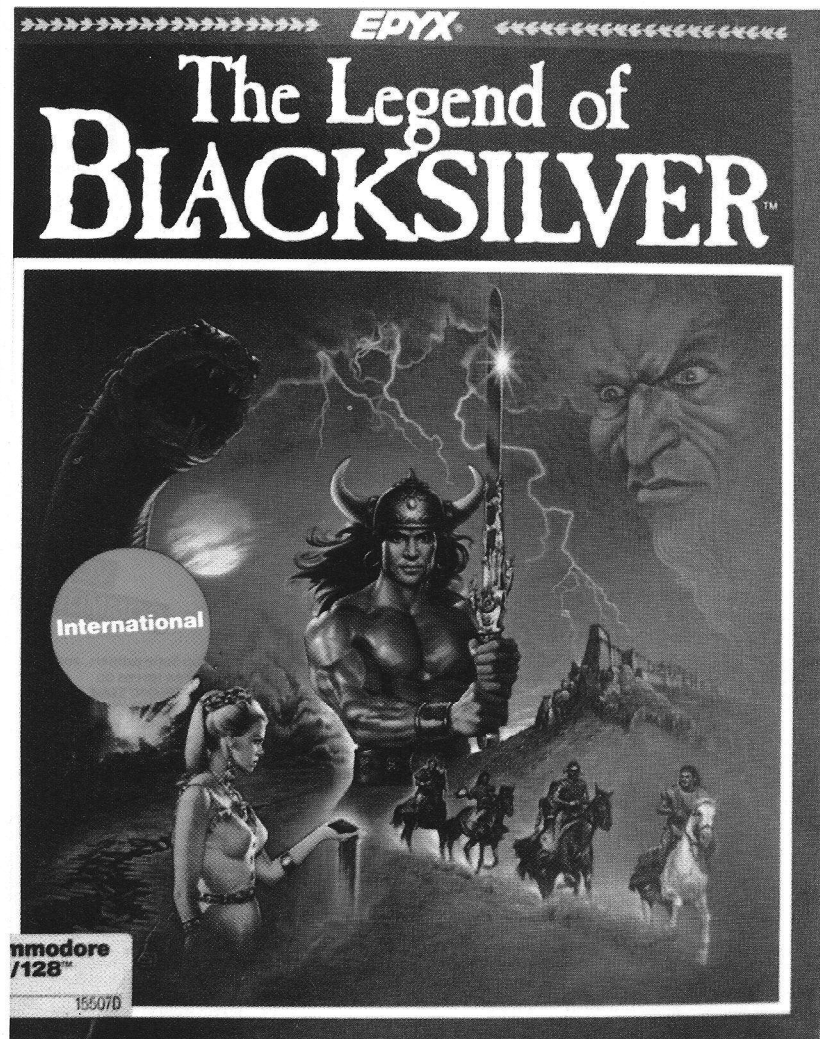
Omdat er 8192 bytes nodig zijn om een hires scherm op te slaan, zijn er per bank 2 gebieden om het hires in op te slaan. Voor bank 0 zijn dat de gebieden 0-8191 en 8192-16383. De andere drie banks hebben dezelfde tweedeling. Met het vierde bit van adres 53272 kunt u het gebied uitkiezen dat u wilt. Als dit bit uitstaat, komt het hires in het eerste gebied van de bank, als het bit aanstaat in het tweede.

Handboek noodzakelijk!

Adventures uitdaging

Software pakketten uitbrengen speciaal gericht op de Nederlandse markt is en blijft een moeilijke zaak. Geïmporteerde software lijkt veel makkelijker maar dat is meestal maar schijn. Pakketten zijn veelal in gekraakte versie veel eerder op de Nederlandse markt 'verkrijgbaar'. De spelletjes, dat is de meest gekopieerde software, hebben meestal geen handleiding nodig. Een ieder die een beetje thuis is op de computer snapt al snel hoe het spel gespeeld moet worden.

Wat anders ligt dat met de serieuze software, hier is de handleiding veel meer van belang. En dan hebben we nog een catagorie waar het spelen van het spel vrijwel onmogelijk is zonder een goede handleiding. We praten hier dan over de adventures. Dit soort spelen zijn zijn meestal beeldverhalen waar de speler een grote inbreng in heeft. Er blijkt in Nederland ook een grote groep mensen te zijn die niets liever spelen dan deze (Engelstalige) adventures. Zij zien het als een uitdaging zich helemaal aan zo'n spel te wijden. Het kan weken, ja zelfs maanden duren, als het je al lukt, om het spel te doorgronden. Sommige komen er nooit doorheen, anderen die meer ervaring in dit soort spelen hebben, hebben er beduidend minder moeite mee.



The legend of the Blacksilver

Legend of BlackSilver is geen uitzondering en heeft een redelijk dikke handleiding. Hierbij is een landkaart op poster formaat gevoegd. Het geheel lijkt op een ouderwetse op perkament gedrukte schatkaart. Dit is een onmisbaar instrument.
Het spel start met een stukje sprookje en is genoemd "een roep naar avontuur". In het kort komt het op het volgende neer:

Ik ben niet langer een simpele ziel. Sinds mijn zeer speciale droom, is er niets meer hetzelfde. Het was verbazend maar waar. Prinses Aylea, dochter van de koning van alle Thalen, is speciaal naar me toe gekomen. We hebben een held nodig, fluisterde de prinses. We hebben een strijder nodig waar alle anderen faalden. In mijn droom heb ik steeds gevraagd, 'waarom ik', maar ze vertelde me: 'slaap rustig verder terwijl ik het gehele verhaal vertel'.
Baron Taragas heeft het Blacksilver, het zwarte zilver, in zijn mijn gevonden, en nu vrezen we voor de veiligheid van ons allemaal. Het heeft de kracht om bergen op te werpen en de grote zeëen te dempen. De krachten om het land te vernietigen. Mijn vader heeft een groot leger bij elkaar geroepen om tegen Taragas ten strijde te trekken. Maar helaas is hij ontvoerd voor hij de legers kon gaan aanvoeren. De tovenaar Seravol heeft me verteld dat het leger op deze manier nooit de strijd kan leiden. Er moet één persoon, een speciale strijder zijn. Na nog een paar woorden gesproken te hebben, kreeg ik een veer in mijn handen gestopt en verdween in dromenland.
Ik bekijk de wereld nu met de ogen van

een strijder. Ik ben gestorven en herboren. Het enige dat mij bijgebleven is zijn de laatste woorden van de prinses, Ik had iedereen kunnen kiezen maar ik koos jou, Ik mag dus niet falen...

Op deze manier wordt het spel ingeleid. Het spel kan met het toetsenbord maar ook met een joystick worden gespeeld. Om verder te komen moet U steeds kiezen uit een groot aantal korte commando's. Zij kunnen uit een lijst worden gekozen met de joystick, maar sneller gaat het door de eerste letter van het commando in te tikkken. Bij het starten van het spel bevindt je je aan de rand van het land van Thalen. Er is een grote verscheidenheid aan landschappen. Het Xamine commando geeft een suggestie wat je het beste bij een bepaald landschap kunt doen. De kaarten werden in die tijd niet zo nauwkeurig getekend als tegenwoordig. Er wordt bij het programma een aantal losse stickers bijgeleverd, waarmee de verschillende onderdelen op de kaart nauwkeurig kunnen worden aangegeven. Er zijn 64 verschillende soorten tegenstanders die je op de route kunt tegenkomen. Ze zijn grof te spliten in twee types, monsters en Humanoids. Monsters zijn altijd vijandig, humanoids kunnen heel vriendelijk zijn, het blijft dus opletten. In de steden, er zijn 16 verschillende steden, zijn de noodzakelijke winkels en banken, zonder hen is een overwinning niet mogelijk. Maar ze kunnen ook een ondergang betekenen. In de tempels bevindt zich een gamemaster, maar ook hier geldt, het is niet allemaal wat het lijkt, voorzichtigheid is de enigste mogelijheid om te overleven.
Achter in de handleiding is een volledig overzicht opgenomen van personages die in het spel voorkomen. Om het spel tot een goed einde te brengen is het noodzakelijk om alle gegevens die hierin vermeld worden goed door te lezen. De persoonlijke eigenschappen van een tegenstander kennen kan bij een strijd de doorslag geven. Eén troost wij hebben het spel niet tot het einde kunnen spelen. Lukt het U wel, dan horen we dit graag van U, maar dan met een goede beschrijving zodat een ander er ook wat aan heeft.



Pool of Radiance

Pool of Radiance is de eerste uit een reeks fantasy rollenspel adventures. Zij komen allemaal van de hand van de legendarische producer van *Advanced Dungeons & Dragons*. De computergraphics zijn kleine kunstwerkjes op zich. Van ieder karakter is er een volledig portret op het scherm. De steden worden in een realistisch 3-D beeld weergegeven.
Het spel speelt zich af in een tot ruïne geworden stad in het noordelijk deel van de Moonsea. Hier proberen avonturiers van een ontwikkelde natie de stad te herbouwen. Vier stukken gereedschap kunnen je hierbij helpen. Als eerste is er een regelboek, hierin zijn vele omschrijvingen te vinden. Het volgende is het avonturen boek. Hierin zijn verhalen opgenomen en vele tekeningen. Dit is een onmisbaar stuk gereedschap, zonder dit onderdeel goed bestudeerd te hebben is het spel zeker niet te spelen. De snel-startkaart, hier wordt uitgelegd hoe het spel te spelen is. En als laatste maar zeker niet het belangrijkste onderdeel het vertaal-wiel. Dit wiel is noodzakelijk om de tekens in leesbare tekst om te zetten.

Het verhaal draait om een locatie in het noordelijk deel van de Moonsea in een reeds vergeten koninkrijk. Phlan was de eerste grote stad van de Moonsea, enkele duizenden jaren geleden. Deze sprookjesachtige stad is ingenomen door een groep monsters. Zij staan onder leiding van een mysterieuze leider. De opdracht luidt dan ook ontmantel deze boosdoener en redt de stad van de ondergang.
Het spel is alleen maar te begrijpen als de history van het verhaal bekend is. In het begeleidende journaal wordt deze geschiedenis volledig verteld, inclusief brieven, schetsen en sugesties. Het verhaal is geheel in stijl gedrukt in een oud Engelse letter. U moet redelijk goed thuis zijn in de Engelse taal om dit spel te kunnen spelen.
De programma's worden door HomeSoft Benelux in Nederland geïmporteerd. Ze zijn verkrijgbaar bij alle goede computerzaken. Bij deze pakketten zijn boeken met tips verkrijgbaar, om het spel uit te kunnen spelen. In de doos bij de software is hiervoor een speciale bestelkaart meegeleverd.

OUD VAN GOUD

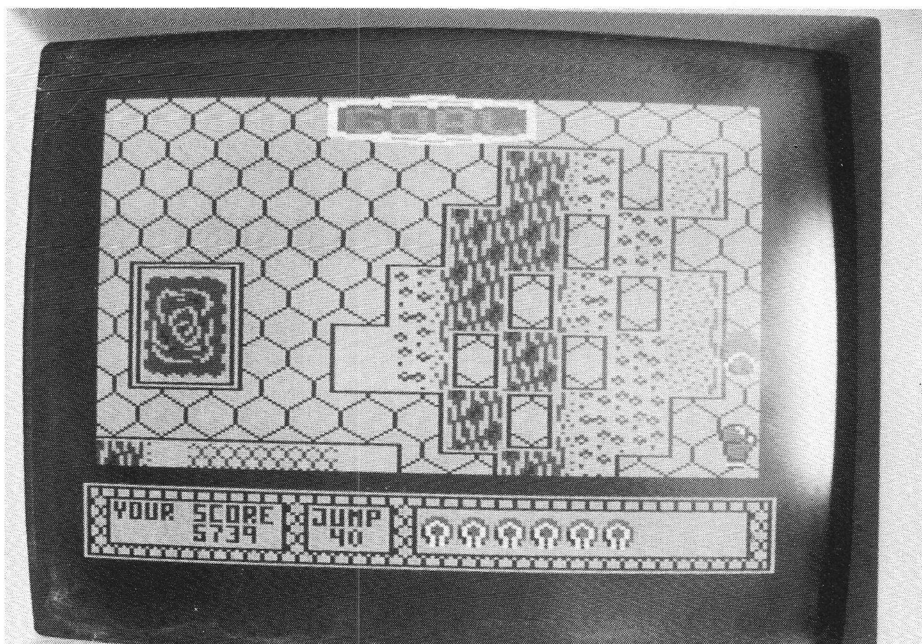
Dynamite Dan

De boosdoeners in dit (moeilijke) spel zijn Dokter Blitzer en zijn aantrekkelijke assistente Donna. Wat is de oorzaak van alle problemen? Dr. Blitzer heeft de plannen voor een "SUPER PSYCHON MEGA RAY" in zijn bezit. Deze plannen zijn verstopt in een geheime kluis in zijn huis. Deze twee personages zijn vals en gemeen, onze held, dynamite Dan is juist het tegengestelde. Hij is dan ook GOUD-en GOUDeerlijk en komt juist op voor de gewone man. Hij is een topagent bij de geheime dienst en inzetbaar bij elke klus. Dan is door zijn slimheid uitgekozen om de opdracht tot een goed einde te brengen.

De opdracht luidt: maak het Dr. Blitzer moeilijk en redt de geheime papieren uit de kluis. Denk erom, het is zeker geen gemakkelijke klus. Vliegen met een zeppelin valt niet mee, maar om deze zeppelin ook nog op het dak van Dr. Blitzer te laten landen, dat is extra moeilijk. De papieren zitten goed opgeborgen in een kluis, om ze eruit te halen moet de kluis opgeblazen worden. Hiervoor is dynamiet noodzakelijk. De acht staven die hiervoor nodig zijn zijn merkwaardige genoeg ook in het huis verborgen. Deze moeten dus eerst opgezocht worden. We zeiden het al, Dr. Blitzer is erg gemeen, hij heeft de staven dan ook verborgen door zijn gehele huis. Natuurlijk liggen deze niet zomaar voor het oprapen, het spel zou dan snel afgelopen zijn. Er is een groot aantal hindernissen te nemen.
Het begint al meteen goed, meteen bij binnenkomst blijkt al dat er een groot aantal diertjes over de grond kruipt. Raketten schieten snel op en neer, we hoeven het er haast niet bij te vertellen, maar aanraken is natuurlijk niet de bedoeling. Om dit gekrioel even rustig te bekijken en te beoordelen wat de beste richting is kun je op de "P", van pauze, drukken. Welke route het beste is willen we je niet vertellen, hoe het spel eindigt kunnen we je niet vertellen (we zijn niet zover gekomen). Misschien zijn er zelfs wel lezers die een paar handige tips hiervoor hebben. We kunnen op onze route ook nog verplaatsmachine's tegenkomen. Deze hokjes verplaatsen ons een eind verder. Dit is niet altijd even handig, maar daar kom je vanzelf wel achter. De eerste verplaatsmachine kun je al vanuit de zeppelin zien, de zeppelin rechts verlaten, de trap op links en je bent er. De bedoeling is om onder de stralendouche te gaan staan en je wordt automatisch verplaatst. Alleen ervaring kan hierbij een goede leidraad zijn. De lift kan je helemaal naar de kelder van het gebouw brengen. Het blijkt dat je hier bij een ondergrondse rivier bent aangekomen. Er drijft een kistje op de rivier, hiervan is gebruik te maken door je er op te laten vallen. Je drijft nu met de rivier mee. Dit gaat niet vanzelf, je moet lopen om je in evenwicht te houden, anders is een nat pak het resultaat. Ook in deze kelder blijkt al gauw dat het in dit huis niet helemaal pluis is. De energievoorraad die je bezit is alleen maar op peil te houden door regelmatig te eten. Naast energie levert dit ook nog punten op. Soep, 40 punten, een kopje thee, 35 punten en een eitje maar 5 punten.
Behalve de etenswaren is er ook nog een groot aantal non-food materialen die punten opleveren, zoals deodorant en creditcards. Alles wat kruipt, vliegt of zelfs maar beweegt is dodelijk, denk daarom. De trampolines moeten gebruikt worden om voorwerpen die zich op een te hoog niveau bevinden te pakken. De veren zijn hier ook voor te gebruiken, maar dit is veel gevaarlijker. Levensgevaarlijk zijn ook de gammastralen. Zij zijn maar enkele momenten aan en daardoor juist zo gevaarlijk. Maak hiervan een situatieschets, zo kom je minder snel voor verrassingen te staan en is de weg makkelijker terug te vinden. De kluis is zo gevonden, maar alle acht de dynamietstaven moeten eerst zijn gevonden!!
Ergens wacht Dr. Blitzer op je in een geheime kamer, hij blijft hier op je wachten om je, zodra je levens allemaal zijn verspeeld, het genadeschot te geven met zijn speciale straalpistool. Ondanks alle gevaren toch veel spelplezier gewenst.

Bounder

In dit spel ben je een tennisbal, deze ligt bovenop het dak van een torenhoge flat. Zodra het spel is gestart begint het beeld te scrollen. Je moet nu wel meebewegen wil je niet vroegtijdig het loodje leggen. Het is zaak goed te kijken welke kant je op moet sturen. Je stuurt o zo snel een doodlopende straat in. Het lukt dan lang niet altijd om over de afgrond heen te springen. Op deze manier ben je gauw door je levens heen.
Gebruik vooral de pijltoetsen om grote afstanden te overbruggen. Onderweg kun je vraagtekens tegenkomen. Kijk hiermee uit, vraagtekens zijn en blijven raadsels. Vaak kunnen ze je een voordeel opleveren. Maar er zijn er ook, voorzien van een kunstgebit, die er een groot genoegen in scheppen om de tennisbal tot moes te vermalen. In het eerste level blijken de moeilijkheden nog wel mee te vallen, een beetje oefenen in het springen en level twee

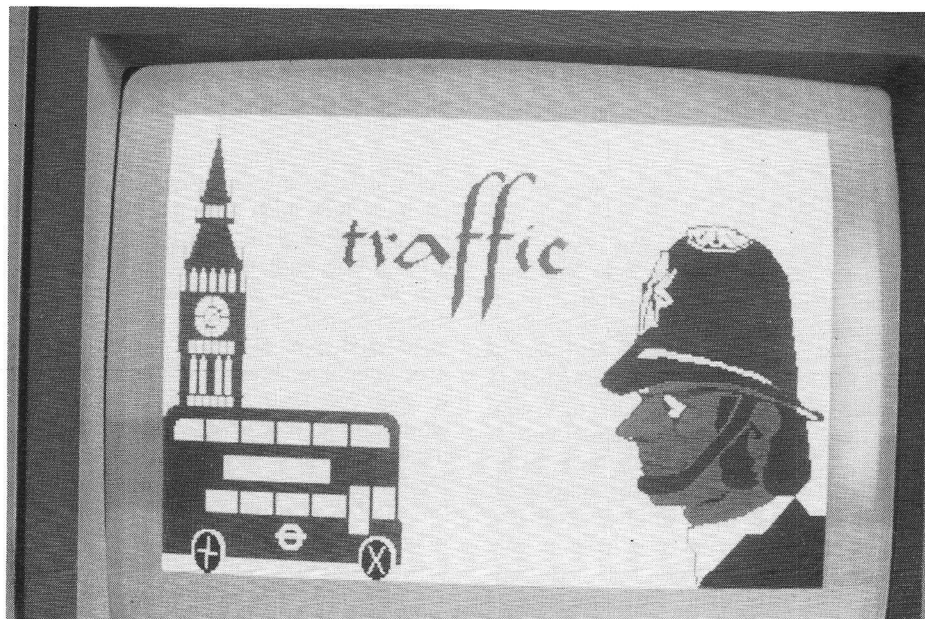


Bounder

is de beloning. Een geoefende speler lacht om dit eerste level.
In het tweede level zwermen er een aantal dikke wespen rond, probeer deze te vangen, ze leveren een groot aantal bonuspunten op. Na elk volbracht level wacht er een bonusronde. Onderschat deze ronde niet, het kan heel belangrijk zijn. Hierin is het namelijk mogelijk het aantal vraagtekens te verminderen. Dit gebeurt door erop te landen. Buiten dit voordeel levert het ook nog punten op. In elk level worden de vijanden vijandiger, de daken smaller en vliegen er meer dingen in de rondte. Er zijn zelfs vraagtekens die verbonden zijn met een pijltjesgooier, pas dus goed op. Het derde level wordt grotendeels beheerst door een lachende tegenstander. Deze is niet te missen, je komt hem al in het begin tegen. Zorg dat jij het bent die het laatste lacht en ga hem maar uit de weg. Ook hier komen we de reeds bekende pijltjesgooier tegen, alleen is hierbij het vraagteken niet meer nodig. De hitte maakt ventilatoren noodzakelijk. Zij zijn zo sterk dat je erdoor weg wordt geblazen, maar met wat aandacht is dit ook in je voordeel te gebruiken. Sommige tegels in dit level zijn niet wat het lijkt. Onbekende vijanden proberen de vrije loop op alle manieren te verhinderen, hou hiervoor ook het luchtruim goed in de gaten. Er zijn in het totaal maar liefst tien verschillende levels. De bergen worden steeds moeilijker om te nemen, vijanden blijken elke keer weer wat slimmer en talrijker te worden. Water blijkt een grotere hindernis dan het lijkt. Het is een moeilijke, zo niet onmogelijke opdracht om het spel helemaal uit te spelen. Tips zijn er eigenlijk nauwelijks te geven, de beste raadgever is ook hier de ervaring. Alleen maar door te proberen kom je erachter wat een bepaald vraagteken nu precies inhoudt. Probeer hierbij aan het wijze spreekwoord te denken: Een ezel stoot zich in 't gemeen ...... (of wel soms). Aan het eind wordt het wel erg moeilijk als je in een veld belandt dat vol staat met de verraderlijke vraagtekens. Pas hier extra op, er schuilt een addertje onder het gras, of eigenlijk onder de vraagtekens!!
Dit spel is er weer één van het soort: heeft het je eenmaal te pakken dan raak je eraan verslaafd. Uren zijn er dan achter de computer door te brengen, dit soms tot ongenoegen van de andere huisgenoten, dus voorzichtig met deze software.

Traffic

Zoals de naam al doet vermoeden heeft het verkeer een groot aandeel in dit spel. Verkeerslichten leiden het verkeer in goede banen. Worden deze lichten verkeerd bediend dan leidt dit zeker tot een verkeerschaos. U kunt zich voorstellen wat er gebeurt in Londen als er zich problemen met deze lichten voordoen. En dat is in dit spel het geval. In dit programma ligt bij jou de taak om de verkeersstroom te beïnvloeden door op een juiste manier de verkeerslichten te bedienen. Deze verantwoordelijke baan staat je te wachten. Je bent de enige die de stad van de ondergang kan redden. Je staat er echt helemaal alleen voor, ook de anders zo vriendelijke bobby's zijn door een virus uitgeschakeld en zijn thuis in hun warme bed gebleven. Na je jarenlang aan het verkeer te hebben geërgerd en gemopperd te hebben op die .... verkeerslichten. Hier is dan je kans, kijk maar wat jij er van kan maken. Zodra het spel is opgestart zie je op het beeldscherm een deel van het stratenplan van Londen. Alles lijkt heel rustig, maar hou je vast, het verkeer komt eraan en in grote aantallen. Auto's, bussen, alles begint te rijden. Het is makkelijk om door een druk op de run/stop toets het beeld te bevriezen. Je kunt nu op je gemak het geheel bekijken en de beste beslissing nemen. Het spel gaat weer verder door de spatiebalk in te drukken. Het verkeersgeruis is dan duidelijk hoorbaar. Op één van de kruispunten zie je een kruisvizier staan. Dit wil zeggen dat dit het kruispunt is waar je de verkeerslichten kunt bedienen. Het verkeer dat rood licht heeft wacht netjes en verkeer met een groen licht trekt langzaam op. Nu is het de bedoeling stuk



Traffic

voor stuk op de kruispunten de lichten te wijzigen zodat de verkeersstroom op gang blijft. Hierbij is het van belang dat er nergens lange files ontstaan. Zodra dit het geval is zie je bij Q een cijfer staan. Dit cijfer geeft het aantal files in dit stadsdeel aan. Zodra de automobilisten ongeduldig beginnen te toeteren dan worden de files te lang. Laat deze files niet te lang worden want dan is het spel snel afgedaan als een hopeloze zaak. Vergeet ook niet dat je je in Engeland bevindt, ze doen hier alles anders en ze rijden ook nog links. Dit is zeker ook even wennen. Als een auto van richting wil veranderen kun je dit zien aan de richtingaanwijzers. Let hier goed op, er kunnen hierdoor lange files ontstaan, want heb je net het verkeer groen licht gegeven dan kan deze auto de boel lelijk blokkeren en alles komt toch nog vast te zitten. De in Nederland 'prima' functionerende groene zone kun je hier ook proberen in te voeren. Wie weet voldoet dit hier wel goed. Houd bussen en de zondagsrijders extra in het oog. Zij willen nog wel eens door het net op rood springende licht rijden, met alle gevolgen van dien. En vergeet bovenal niet dat een Engelse automobilist een heer in het verkeer is, zij wachten op alles. Dit alles is natuurlijk makkelijker gezegd dan gedaan, tenslotte staan de beste stuurlui altijd nog aan de wal, of in dit geval aan de kant. En wie weet geeft het spelen van dit spel ook wat meer inzicht in het echte verkeer, het blijkt dat ook jij zelf het niet helemaal goed kan doen (misschien wel helemaal niet).

Trashman

Vele beroepen zijn al de basis geweest voor computerspelletjes. Natuurlijk zegt dit niet alles over het beroep, vaak lijkt het er in de verste verte niet meer op. Maar dit mag de pret niet drukken. Ditmaal is de vuilnisophaler aan de beurt. Nu niet meteen je schouders ophalen, er zijn wel niet zoveel jongens en meisjes die vuilnis"man" willen worden als piloot of dokter, maar in dit spel wil iedereéén het wel. Ook in dit spel krijgen we weer een bevestiging van de regel, alle begin is moeilijk, zelfs vuilnis ophalen moet worden geleerd. Het spel start in de kortste straat, en dat is maar goed ook, zo kan er wat ervaring worden opgebouwd.
Staunton Road valt de eer te beurt om als eerste van de vuilnis verlost te worden. Er staan hier niet meer dan vijf huizen, en ook niet meer dan vijf bakken die moeten worden geleegd. Denk niet: dat heb ik zo voor elkaar, want dan kom je bedrogen uit. Er moet netjes gewerkt worden, grasvelden mogen niet gebruikt worden. Je wordt soms zelfs keurig gewaarschuwd door een bordje "denk om de hond". Sla deze waarschuwing niet in de wind, deze borden staan er niet voor niets. Gebruik de paden, wijk daar niet van af, alleen op deze manier zijn er voldoende bonuspunten te verkrijgen. De bedoeling is de bak op te halen, deze te legen en dan weer netjes terug te zetten waar deze stond. Het blijkt dat de vuilophalers niet erg geliefd zijn bij de weggebruikers. Dus kom niet te dicht aan de rand als er een auto nadert. Lukt het je om de lege bak weer op zijn oude plaats terug te zetten, vaak weer helemaal achterin de tuin, dan komt de bewoner naar buiten. Deze zal je dan vriendelijk te woord staan en je om een gunst vragen. De bedoeling is dan dat je mee loopt zijn huis binnen. Je krijgt na afloop een fooi. Dat is erg belangrijk, want die wordt later omgezet naar bonuspunten. Zonder deze bonuspunten is het niet mogelijk om een volgende straat af te handelen. Soms heb je het idee dat een vuilnisbak op een plaats staat die niet via een pad te bereiken is. Kijk dan ook achter het huis maar ga in geen geval via het gras. Zodra een straat geheel is afgewerkt loop je naar het einde van de straat om naar de volgende straat te gaan. In deze straat is zelfs de stoep gevaarlijk geworden, hier vliegen jongens rond. Ze rijden je zo van de sokken, met het gevolg dat je je een aantal momenten niet meer kan bewegen. Pubs lijken erg aantrekkelijk, het is hier gezellig en warm, maar deze verleiding is niet van alle gevaar ontbloot. Je ziet: het vak van vuilnisophaler gaat ook niet over rozen. Al met al een gezellig eenvoudig spel, waarbij niet te veel moet worden nagedacht.

RG

Tips & Trucs 64

Prinses lente heeft de strijd aangebonden met koning winter. Het zal niet lang meer duren eer deze verschrikkelijke strijd in het voordeel van de schone prinses beslecht is. Maar voordat zij met haar charme de knoppen aan de bomen doet ontwaken en de vogeltjes doet zingen, bieden Hylke Sprangers en Michel de Boer u wederom een bonte verzameling tips & trucs, die uitnodigen tot een flinke worstelpartij met uw Commodore 64.

In deze aflevering hebben we onder andere de volgende onderwerpen: verbeterde FRE functie, register opslag, joystick besturing en alles over random.

Verbeterde FRE

In de handleiding van de Commodore 64 staat dat met de instructie FRE(X) het aantal bytes, dat vrij is voor Basic, opgevraagd kan worden. Waarbij voor X een willekeurige waarde of een string mag worden ingevuld. Als u echter direct nadat u de computer heeft aangezet de instructie PRINT FRE(0) intikt, krijgt u een negatieve waarde. Deze waarde klopt niet; wat moeten we ons voorstellen bij een negatief aantal bytes? Hoe komt het dat de computer een negatieve waarde retourneert en hoe kunnen we de juiste waarde berekenen?
Dat er een negatieve waarde geretourneerd wordt, is als volgt te verklaren. De machinecode voor de FRE functie berekent hoeveel bytes er vrij zijn en vertaalt het resultaat naar floating point formaat. Bij dat vertalen gaat het fout. Hiervoor wordt namelijk de routine op adres $B391 gebruikt. Deze routine vertaalt twee-complement 16-bits integers naar floating point formaat. Hierbij worden integers waarvan het meest linkse bit aanstaat, opgevat als negatieve getallen (zie vorige aflevering van 'Tips & Trucs 64' voor een uitgebreide beschrijving van twee-complement getallen). Deze routine levert dus waarden die tussen -32768 en 32767 liggen. Bij de voorloper van de Commodore 64, de VIC 20, was dit geen probleem, omdat er altijd minder dan 32768 bytes vrij waren. Bij de Commodore 64 kunnen er meer dan 32767 bytes vrij zijn. Dit aantal wordt dan opgevat als een negatief getal. Dit kan verholpen worden door de machinecode voor de FRE functie zodanig te veranderen, dat niet de vertaalroutine op adres $B391 wordt gebruikt, maar een routine die gewone integers tussen 0 en 65535 vertaalt. Om de code te veranderen moet het Basic ROM eerst naar RAM worden verplaatst. In RAM kunnen we dan de verandering aanbrengen. Het onderstaande programma doet dit.

```
5 rem verbeterde fre
10 for i=679 to 696
20 read a:poke i,a:next
30 for i=40960 to 49151
40 poke i,peek(i):next
50 poke 1,peek(1) and 254
60 for i=45965 to 45967
70 read a:poke i,a:next
100 data 165,52,229,50,162,0,134,13
110 data 133,98,132,99,162,144,56,76
120 data 73,188
130 data 76,167,2
```

In regels 10-20 wordt er een routine die gewone integers vertaalt, op adres 679 gezet. Regels 30-50 verplaatsen het Basic ROM naar RAM. Tenslotte wordt in regels 60-70 de verandering aangebracht. Als u na het runnen van dit programma de FRE functie uitvoert, krijgt u de juiste waarde.

Regsiter opslag adressen	
Adres	Register
780	A register (accumulator)
781	X index register
782	Y index register
783	P register (status)

Figuur 1

Register opslag

De machinetaalprogrammeur heeft op de Commodore 64 vier registers tot zijn beschikking; het A register, X index register, Y index register en P register. Het A register wordt meestal accumulator genoemd. Het P register is het status register. Dit register bevat alle vlaggen, zoals de carry en overflow vlag.
Een machinetaal programma kan vanuit Basic met het SYS commando worden aangeroepen. In het begin van het programma hebben de vier registers een onbekende waarde. Deze registers worden dan in het machinetaal programma geinitialiseerd. (Initialiseren betekent dat aan de registers een beginwaarde wordt gegeven.) Het is echter ook mogelijk om de registers vanuit Basic te initialiseren, zodat de registers vanaf het begin van het machinetaal programma een bekende waarde hebben. Zodra het SYS commando wordt gegeven, wordt namelijk niet meteen gestart met het uitvoeren van het machinetaal programma, maar worden eerst de waarden van de adressen 780-783 in de registers geladen. In figuur 1 staat welk adres in welk register wordt geladen. Om er bijvoorbeeld voor te zorgen dat de accumulator de waarde 5 heeft bij de start van het programma, moet u in adres 780 een 5 poken, voordat u het SYS commando geeft. Nadat het machinetaal programma beeindigd is, worden de waarden van de registers in de adressen 780-783 gestopt, zodat u vanuit Basic kan bekijken welke resultaten het machinetaal programma in de registers heeft gestopt. De adressen 780-783 worden alleen gebruikt bij het SYS commando en niet bij de USR instructie.
Adres 783 behoeft nog enige uitleg. In dit adres correspondeert elk bit met een vlag. In figuur 2 staat welk bit correspondeert met welke vlag. Als bijvoorbeeld voor de aanroep van een routine de carry vlag

Bitbetekenis van adres 783		
Bit	Vlag	Vlagnaam
7	N	Negative
6	V	Overflow
5	-	Niet gebruikt
4	B	Break
3	D	Decimal
2	I	Interrupt Disable
1	Z	Zero
0	C	Carry

Figuur 2.

moet worden aangezet, dan moet u bit 0 van adres 783 aanzetten met bijvoorbeeld

```
POKE 783,PEEK(783) OR 2^0.
```

Een handige toepassing van de adressen 780-783 is het positioneren van de cursor. Dit kan gedaan worden met de ROM routine op adres 65520. Voor de aanroep van deze routine moet het kolomnummer in het Y register en het regelnummer in het X register worden gezet en de carry vlag moet worden uitgezet. Het volgende voorbeeld programma positioneert de cursor op regel 12 en kolom 18 en print op deze positie 'hallo'.

```
10  rem cursor positioneren
20  poke 781,12:poke 782,18
30  poke 783,peek(783) and
     (255-2^0)
40  sys 65520
50  print "hallo"
```

Als voor de aanroep de carry vlag aangezet wordt, dan bevatten de X en Y registers na de uitvoering van de routine de cursor positie. Het volgende programma illustreert dit.

```
10 rem lees cursor positie
20 poke 783,peek(783) or 2^0
30 sys 65520
40 print "regel:";peek(781)
50 print "kolom:";peek(782)
```

Joystick besturing

In veel programma's wordt gebruik gemaakt van de joystick; bijvoorbeeld voor de besturing van een Formule 1 in een race spelletje of voor de besturing van een aanwijspijltje in een menu gestuurd programma. In deze aflevering zullen we uitleggen hoe joystick besturing in eigen programma's gebruikt kan worden.
De Commodore 64 heeft aan de zijkant twee poorten waar joysticks ingeplugd kunnen worden. De voorste poort is poort 1 en de achterste is poort 2. Beide poorten zijn aangesloten op de Complex Interface Adapter chip (CIA). Deze chip regelt de communicatie met randappatuur. De gebruiker kan via een aantal geheugenadressen communiceren met deze chip. De adressen 56320 en 56321 dienen voor de communicatie met de joysticks. Logischer wijs zou u verwachten dat adres 56320 bij poort 1 hoort en 56321 bij poort 2. De Commodore 64 is echter niet zo logisch en adres 56320 hoort dan ook bij poort 2 en 56321 bij poort 1. Als u de joystick in poort 2 beweegt, dan verandert de waarde van adres 56320.

Bitbetekenis van de joystick adressen 56320 en 56321

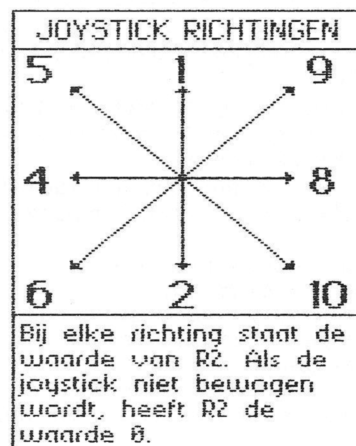
Bit	Richting
0	omhoog
1	omlaag
2	links
3	rechts
4	vuurknop

Figuur 3.

Als u uw joystick uit elkaar haalt, dan kunt u zien dat een joystick uit een hoop draden en vijf schakelaartjes bestaat; vier voor de richtingen links, rechts, omhoog en omlaag en een voor de vuurknop. Sommige joysticks hebben nog een zesde schakelaar voor een extra vuurknop. Deze is echter met dezelfde draadjes als de schakelaar voor de eerste vuurknop verbonden. Elke schakelaar in de joystick correspondeert met een bit in adres 56320 of 56321. Hiervoor worden bits 0-4 gebruikt. Bits 5-7 hebben een andere functie. In figuur 3 kunt u zien welk bit overeenkomt met welke schakelaar. Als een schakelaar geen contact maakt heeft het overeenkomstige bit de waarde 1 en als er wel contact gemaakt wordt heeft het bit de waarde 0. (Ja, de Commodore 64 is wel consequent onlogisch.) Als de joystick niet bewogen wordt, staan alle bits dus aan. Door te testen welke bits er uit staan, kan bepaald worden in welke richting de joystick bewogen wordt. Als de joystick niet in vertikale of horizontale richting bewogen wordt, maar in diagonale, dan maken twee schakelaartjes tegelijk contact en er worden dus twee bits uitgezet. Bits 0 en 2 worden bijvoorbeeld uitgezet als de joystick naar links en omhoog bewogen wordt. Het is nogal onhandig om de vijf bits een voor een te testen met de AND instructie. Met het volgende statement kunt u eenvoudig testen in welke richting joystick 2 wordt bewogen:

```
R2=NOT PEEK(56320) AND 15
```

In figuur 4 staat welke waarde R2 heeft bij de acht verschillende richtingen. De waarden 3 en 7 komen in deze figuur niet voor omdat deze alleen kunnen optreden als de schakelaars voor links en rechts of omhoog en omlaag tegelijk worden ingedrukt. Dit is onmogelijk. U kunt nu eenvoudig met een ON R2 GOSUB construc-



Figuur 4.

Rectificatie Tips & Trucs

In het artikel "Tips & Trucs op de 64" uit de vorige aflevering, is in de listing voor het vergelijken van twee-complement getallen het een en ander weggevallen. We geven daarom deze listing nog een keer, zonder commentaar.

```
LABELS  MNEMONICS
         lda #1
         sta data
         sec
         sbc #-1
         bmi min
         bvs kleiner
         bvc groter
min      bvc kleiner
         bvs groter
kleiner lda #<kl
         ldy #>kl
         jsr $able
         jmp end
groter  lda #<gg
         ldy #>gg
         jsr $able
end      lda data
         rts

data     .byt 0
kl       .asc "accu < getal"
         .byt 0
gg       .asc "accu >= getal"
         .byt 0
```

Nogmaals onze excuses voor de gemaakte fouten.

tie, bij elke richting de juiste actie ondernemen.
Met het volgende statement kunt u testen of de vuurknop van joystick 2 is ingedrukt.

V2=PEEK(56320) AND 16

V2 heeft de waarde 0 als de vuurknop is ingedrukt en 16 als dit niet zo is.
Joystick 1 kan op dezelfde wijze getest worden. Adres 56320 moet dan vervangen worden door adres 56321.
Het gebruik van joystick 1 geeft enkele problemen. Adres 56321 wordt namelijk niet alleen voor de joystick gebruikt maar ook voor het scannen van het toetsenbord. Beweegt u de joystick in poort 1 maar eens naar rechts. Er verschijnt dan een '2' op het scherm. De computer kan geen onderscheid maken tussen het naar rechts bewegen van de joystick en het intikken dan de '2'. U kunt wel voorkomen dat de invoer van het toetsenbord beschouwd wordt als invoer van de joystick, door het scannen van het toetsenbord te stoppen als u de joystick test. Dit kunt u doen met POKE 56333,127:POKE 56321,255. Met POKE 56333,129 kunt u het scannen van het toetsenbord weer starten.
Het is niet mogelijk om de joystick uit te schakelen.
Als u maar een joystick nodig heeft, raden wij u aan om poort 2 te gebruiken. U omzeilt dan alle problemen.
Het volgende programma verduidelijkt het gebruik van de joystick in poort 2:

```
10 rem joystick besturing
20 print "{SHIFT-CLR}";
30 r2=not peek(56320) and 15
40 v2=peek(56320) and 16
50 r$="geen          "
60 on r2 gosub
   120,130,140,150,160,170,180,190,200
   ,210
70 v$="niet ingedrukt"
80 if v2=0 then v$="ingedrukt
   "
90 print "{HOME}richting: ";r$
100 print "vuurknop: ";v$
110 goto 30
120 r$="omhoog          ":return
130 r$="omlaag          ":return
140 rem waarde 3 is onmogelijk
150 r$="links           ":return
160 r$="links omhoog ":return
170 r$="links omlaag ":return
180 rem waarde 7 is onmogelijk
190 r$="rechts          ":return
200 r$="rechts omhoog":return
210 r$="rechts omlaag":return
```

In regel 30-40 worden de waarden voor r2 en v2 berekent. In regel 50-60 wordt de waarde voor r$ bepaalt. r$ bevat de richting waarin de joystick bewogen wordt. U moet opletten dat er in de ON GOSUB constructie ook regelnummers voor de waarden 3 en 7 staan, hoewel deze niet kunnen optreden. Op deze regelnummers kan bijvoorbeeld een REM statement worden geplaatst zoals in regels 140 en 180. In regel 80 wordt getest of de vuurknop is ingedrukt.

Testbeeld generator

In Commodore Info no. 8 hadden we een programma geplaatst dat het beeldscherm verdeelde in een aantal horizontale balken, een zogenaamd testbeeld. Van de heer Douwe A. van der Weij kregen we een programma toegestuurd, dat een verbetering is van het programma van ons. De verbeteringen zitten in de volgorde van de kleuren, en er is bovendien een achtergrondtoon, behorend bij het testbeeld, toegevoegd. De toon is een blokgolf met een frekwentie van 1000 Hz. Hieronder volgt het programma.

```
10 for i= 49152  to 49259
20 read a:c=c+a:poke i,a:next
30 if c<>13398 then print"fout in
   data":end
40 sys 49152
400 data
   169,212,133,252,169,0,133,251
410 data
   168,145,251,200,192,25,208,249
420 data 162,240,142,6,212,162,15,142
430 data 24,212,162,65,142,4,212,162
440 data 7,142,3,212,162,128,142,2
450 data 212,162,66,142,1,212,162,130
460 data 142,0,212,120,141,17,208,169
470 data 1,141,32,208,169,1,141,32
480 data 208,169,7,141,32,208,169,3
490 data 141,32,208,169,5,141,32,208
500 data 169,4,141,32,208,169,2,141
510 data 32,208,169,6,141,32,208,169
520 data 0,141,32,208,169,0,141,32
530 data 208,76,55,192
```

Random

We zullen in deze aflevering van Tips & Trucs wat extra aandacht besteden aan Random. We zullen uitleggen hoe een computer een 'willekeurige' reeks getallen kan genereren.
De eerste vraag die zich gelijk opwerpt, is uiteraard wat random nou precies is of betekent. Wel, random is afgeleid van het franse woord randir, dat galopperen betekent. In het nederlands betekent random zoiets als willekeurig of toeval. Als we het over computers hebben, dan wordt random meestal geassocieerd met een willekeurige rij getallen. De encyclopedie omschrijft toeval als het niet aanwijsbare van oorzaken, het ongedetermineerde, het onvoorspelbare.
Vraag twee is dan meteen of een willekeurige rij getallen wel te genereren is door de computer, en of toeval eigenlijk wel bestaat. We gaan het hier verder niet hebben over de vraag of toeval bestaat. (Filosofen kunnen over deze vraag uren en uren discussieren, zonder ook maar een stap dichter bij het antwoord te komen, maar dit terzijde). Over de vraag of een willekeurige rij getallen door de computer te genereren valt, kunnen we echter wel kort zijn, dit kan niet. We kunnen de computer echter wel een rij getallen laten genereren, die lijkt op een willekeurige rij. Deze rij getallen zal echter altijd reproduceerbaar zijn. Tevens zal ook steeds elk volgend getal in de rij te voorspellen zijn. Omdat zo'n rij getallen dus niet een echte willekeurige rij is, noemen we zo'n rij pseudo random.
In de Commodore 64 zit de funktie RND waarmee pseudo random getallen kunnen worden verkregen. De computer berekent de getallen aan de hand van een formule. Deze formule zit gewoon standaard in het ROM, en vandaar dat de rij getallen die met de formule wordt verkregen, reproduceerbaar is. Wat we echter wel aan de formule kunnen veranderen, is de beginwaarde. Deze beginwaarde wordt ook wel de seed value (zaad of kiem) genoemd. Door steeds een andere seed te kiezen, krijgen we ook een andere reeks getallen. We gaan nu kijken hoe de funktie RND werkt. Aan RND moet een getal worden meegegeven tussen haakjes. Aan de hand van dit getal wordt een pseudo random getal teruggegeven. Bijvoorbeeld:

PRINT RND(0)

Als u dit invoert, wordt er een getal op het scherm geprint. Het getal tussen de haakjes bij RND is zeer belangrijk. Met een negatief getal tussen de haakjes kan de seed worden veranderd. We geven een voorbeeld:

PRINT RND(-2)

Op het scherm zal nu de waarde 2.99205567E-08 verschijnen. Deze waarde krijgt u altijd als u RND(-2) intypt. Zo heeft ieder negatief getal haar eigen waarde met RND. Dit zijn de seed values. U kunt een willekeurige seed uitkiezen. De uikomst kunt u bijvoorbeeld in een variabele stoppen die u niet gebruikt, bijvoorbeeld D=RND(-4).
Zoals al gezegd, wordt er, door een andere seed te nemen, een andere reeks getallen gegenereerd door de Commodore 64. Nu de seed is uitgekozen, kunnen we echte pseudo random getallen van de computer krijgen. Dit moet gebeuren met een positief getal tussen de haakjes. Welk positief getal u nu neemt maakt niet uit. De seed value is immers doorgegeven aan de computer, en deze genereert nu gewoon een lijst met getallen. Het enige wat belangrijk is, is dat het getal positief is. We geven nu een klein voorbeeld programma:

```
10 a=rnd(-2)
20 for x=0 to 3
30 print rnd(1)
40 next
```

Als u dit programma runt, zullen de volgende getallen steeds op het beeld verschijnen:

```
0.865554613
```

```
0.846128798
0.981100118
0.986081582
```

U ziet het, de funktie RND genereert altijd getallen die tussen 0 en 1 in liggen. Maar het is natuurlijk niet leuk (en in games zeker niet handig) dat zo steeds dezelfde rij getallen op het beeld verschijnt. Wat we dus willen is een seed die steeds verandert. Immers, als we steeds een andere seed hebben, wordt er steeds een andere reeks getallen gegenereerd. Dit kunnen we bereiken door met de interne klok van de Commodore TI steeds andere seed values te maken. Het bovenstaande programma komt er dan als volgt uit te zien:

```
10 a=rnd(-ti)
20 for x=0 to 3
30 print rnd(1)
40 next
```

Als we het bovenstaande programma runnen, krijgen we steeds een andere rij getallen. Regel 10 zouden we ook kunnen veranderen in a=rnd(0). Dit is namelijk precies hetzelfde als a=rnd(-ti). Als het getal 0 aan RND wordt meegegeven, genereert de computer zelf een seed, aan de hand van de interne klok.

We hebben al gezegd dat RND een getal tussen de 0 en 1 geeft. Als we gehele waarden willen hebben tussen bijvoorbeeld de 0 en de 6, moeten we het volgende intypen:

```
PRINT INT(RND(1)*6))
```

Let wel, het getal 6 zelf zal nu nooit voorkomen en het getal 0 wel. Als we echt waarden van 1 t/m 6 willen hebben, moeten we nog 1 bij de gegeven waarden optellen.

Een Basic programmeur kan gebruik maken van de hierboven beschreven RND funktie. In machinetaal is het veel lastiger om pseudo random getallen te maken. In Commodore Info no.3, april/mei 1989 hebben we reeds uitgelegd hoe in machinetaal een dergelijke reeks getallen kan worden gemaakt. We zullen hieronder deze methode kort herhalen.

In machinetaal kan natuurlijk de ROM routine van RND worden aangeroepen, maar er is een makkelijkere en snellere manier. Er kan een rij getallen via de geluidschip van de Commodore worden gegenereerd. De golfvorm van geluidsregister kan namelijk worden uitgelezen in adres 54299. Als we nu ruis als golfvorm nemen, kan dit adres gebruikt worden als pseudo random generator. De frekwentie van de ruis moet hoog worden gekozen voor een goede werking van dit adres als een pseudo random generator. De ADSR en het volume hoeven overigens niet te worden ingesteld, omdat deze geen invloed hebben op de golfvorm. De getallen die op deze manier worden gegenereerd, liggen uiteraard tussen de 0 en 255. We geven een voorbeeld programma in assembly:

```
Mnemonics Kommentaar
lda $d418
ora #$80
sta $d418 ;schakel stem 3 uit
lda #$ff
sta $d40f ;hoge frekwentie
lda #$80
sta $d412 ;ruis
rts
```

Na het uitvoeren van bovenstaand programma, kan adres 54299 worden uitgelezen voor het verkrijgen van pseudo random getallen. Dit kan gedaan worden met lda $d41b. Overigens wordt in de eerste 3 regels code stem 3 uitgeschakeld, zodat stem 1 en 2 nog gewoon te gebruiken zijn voor muziek. Dit uitschakelen van stem 3 wordt bereikt door bit 7 van adres 54296 aan te zetten.

We hebben al opgemerkt dat bij een aanroep van de funktie RND, de computer niets anders doet dan het invullen van een formule. De uitkomst van de formule is dan zogenaamd een willekeurig getal. Door de formule goed te kiezen, zullen de getallen uniform verdeeld zijn over het interval 0 tot 1. Dit betekent dat als we de getallen bijvoorbeeld omrekenen naar getallen van 0 tot 5, dat de 1 relatief even vaak voorkomt in de reeks gegenereerde getallen als de 0, 2, 3, 4 en 5. Dit in vergelijking met het opwerpen van een dobbelsteen. Als we 100 keer een dobbelsteen gooien, zal blijken dat de 2 ongeveer even vaak bovenkomt als de 1, 3, 4, 5 en 6. Daarbij moeten we wel even aantekenen dat het gooien van een dobbelsteen geen toevalsproces is. Als we alle faktoren zouden kennen (zoals kracht van het opwerpen, wrijving van de lucht, veerkracht van de tafel etc.), dan zou precies voorspeld kunnen worden wat de uitkomst zal zijn.

We keren nog even terug naar de formule in de computer. We kunnen de lijst getallen die de computer genereert, beïnvloeden door steeds andere beginwaarden, seed values, te nemen. Voor het genereren van de lijst getallen zelf wordt vaak de lineaire congruentie methode gebruikt. Het volgende getal in de reeks wordt dan berekend met behulp van de volgende formule:

```
X(I+1) = (A*X(I) modulo C) + B
```

Daarbij zijn A, B en C constanten. Verder levert (X modulo Y) de rest van de gehele deling X/Y. In Basic is (X modulo Y): X - INT(X/Y)*Y. Er moet ook een seed worden gekozen. We stellen X(0) gelijk aan de seed. X(1) kan dan met bovenstaande formule uit X(0) worden berekend. We geven nu een programma die de formule implementeert.

```
5 rem pseudo random 1
6 rem formule: x(i+1)=(121*x(i)
    modulo 99) + 1
10 s=7
20 for x=0 to 9
30 gosub 1000:print r
40 next:end
999 rem subroutine voor getal
1000 a=int((s*121)/99)*99
1010 s=s*121-a+1
1020 r=k/100:return
```

In regel 10 wordt de seed value gedefinieerd. Na een aanroep van de subroutine op regel 1000, bevat de variabele r een getal tussen 0 en 1. Het bovenstaande programma print 10 pseudo random getallen op het scherm. Het nadeel van het bovenstaande programma is, dat de reeks getallen die gemaakt wordt al gauw repeteert. Het volgende programma heeft een complexere formule, waardoor er een betere rij getallen gegenereerd wordt.

```
5 rem pseudo random 2 *************
10 s=7:gosub 100:rem init
20 for i=0 to 100
30 gosub 200:rem bereken getal
40 print r;:next:end
99 rem init ************************
100 dim a(54):a(0)=s
110 ri=0:for i=1 to 54
120 p1=int(31/100):p0=31-p1*100
130 q1=int((a(i-1)+1)/100):q0=(a(i-1)
    +1)-q1*100
140 j1=p0*q1+p1*q0
150 j1=(j1-int(j1/100)*100)*100+p0*q0
160 a(i)=j1-int(j1/10000)*10000:next
199 rem bereken getal
    *****************
200 ri=ri+1-int((ri+1)/55)*55
210 p1=ri+23-int((ri+23)/55)*55
220 q1=ri+54-int((ri+54)/55)*55
230 a(ri)=a(p1)+a(q1)-int((a(p1)
    +a(q1))/10000)*10000
240 r=a(ri)/10000:return
```

Het bovenstaande programma bevat twee subroutines. De eerste begint op regel 100. Deze moet een keer worden aangeroepen, en initialiseert een array met pseudo random getallen. Daarbij wordt de seed value s gebruikt. Telkens na een aanroep van de tweede subroutine, die op regel 200 begint, bevat de variabele r een pseudo random getal.

Als laatste geven we nog de adressen waar de seed value van de RND funktie wordt opgeslagen. Dit zijn de adressen 139-143.

Alle voorbeelden die we hebben gegeven om een reeks willekeurige getallen te verkrijgen, waren geen echte random generators. Overigens, als de quantum ruis van een transistor wordt versterkt en gedigitaliseerd, dan garanderen de huidige wetten van de quantum mechanica, dat de string van nullen en enen niet reproduceerbaar en voorspelbaar is. De vraag is of dat zo blijft.

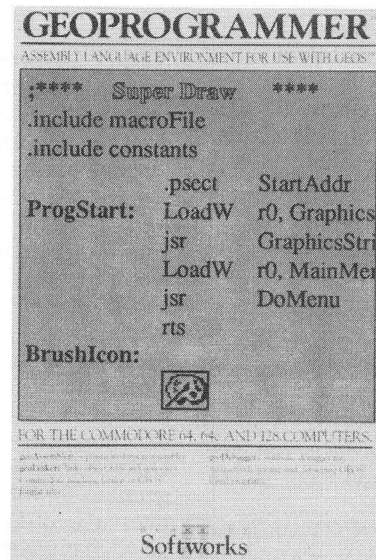
Hylke Sprangers & Michel de Boer

GEOS-machinetaal (9)

Al meer dan een jaar loopt Peter Boncz, cursus over de machinetaal van GEOS. De negende aflevering zal net als de tiende gaan over de hulpmiddelen die de programmeur ten dienste staan.

Aflevering negen en tien zijn geheel gewijd aan de hulpmiddelen die GEOS de programmeur biedt, om zelf programma's voor de GEOS-omgeving ('environment') te ontwikkelen en te onderhouden. Deze hulpmiddelen worden verstrekt door het GeoProgrammerp-akket, inmiddels al ruim meer dan een jaar in omloop. Toen het pakket uitkwam is er ook een bespreking over verschenen in de CI van de hand van Bert Venema. Deze beschrijving was, met respect gezegd, voor de programmeurs onder de lezers nogal oppervlakkig. Het lijkt mij dan ook zeker niet dubbelop om eens diepere aandacht aan GeoProgrammer te schenken, te meer omdat dit een onmisbaar hulpmiddel voor lezers van deze cursus is geworden.

Het GeoProgrammer-pakket bestaat uit drie delen: de GeoAssembler, de GeoLinker en de GeDebugger. De GeoAssembler is het hart van het systeem: het bevat de assembler met enkele bijgeleverde standaard-files. Een assembler is een programma dat assembly vertaalt naar machinecode. Assembly op zijn beurt is machinetaal, gemengd met symbolische namen voor adressen (die door de gebruiker zelf gedefinieerd kunnen worden) en (ook zelf gedefinieerde) macro's. Als een assembly-programma door een assembler heengehaald wordt, dan ontstaat er de zogenaamde 'relocatable'-code (ook wel .rel - code). De naam 'relocatable' zegt het al: het programma kan van plaats in het geheugen veranderd worden en staat nog nergens 'vast'. De Linker (to link = aan elkaar koppelen) heeft tot taak verschillende stukken .rel-code op de juiste wijze in elkaar te voegen en op de goede plaats in het geheugen te zetten. De Linker maakt van verschillende .rel-files een uiteindelijke - uitvoerbare- file. Hiervoor wordt gebruik gemaakt van een speciale link-file, waarin de programmeur vertelt welke .rel-files op welke manier aan elkaar gelinkt moeten worden. Dit hele proces hebben we al in de vorige afleveringen van de cursus gezien: doorgaans was er 1 assembly-file, en 1 header-file, die beide door de GeoAssembler moesten. Vervolgens was er steeds een Link-file, waarin de namen van de .rel files stonden, en de geheugenplaatsen waar ze terecht moesten komen. Nog geen echt nieuws onder de zon dus.



Het derde programma, de GeoDebugger, is tot nu toe buiten beeld gebleven, en dat is ook logisch, want de hier geLISTe programma's horen al te werken (?). Bij Berkely Softworks gaan ze uit van de 'assemble-link-crash-debug'-cyclus voor programma-ontwikkeling: assembleren, linken, testen (=crashen), en repareren met de debugger. Omdat, zoals gezegd, in de cursus het gebruik van de GeoDebugger niet aan de orde is, zal ik hier slechts zeer summier op ingaan, en dat in de volgende aflevering. In deze aflevering zal ik me uitsluitend bezighouden met de GeoAssembler.

GeoAssembler

De GeoAssembler is één van de meest uitgebreide assemblers die voor de commodore 64 beschikbaar zijn. Hij assembleert de assembly-code in 2 stappen (passes). In de eerste stap worden alle namen (labels) die opduiken in het programma in een tabel gezet. In de tweede stap begint de computer van voren af aan, en kunnen evt. foutboodschappen gegeven worden als bepaalde waarden verkeerd gebruikt worden. Hierbij interessant te vermelden is de interne variabele 'Pass1', die waar (1) is als de assembler met de eerste stap bezig is, en onwaar (0) is als de assembler met de tweede stap bezig is. Geen paniek: later zal ik een poging doen om duidelijk te maken wat dit alles te betekenen heeft.

De normale handelwijze van een programmeur is als volgt: eerst wordt het programma in GeoWrite geschreven. Pas als het af is, start men de GeoAssembler op en wordt de GeoWrite-file geassembleerd. Naast de gewone C-64 opcodes; kreten als 'lda', 'sta' en 'jsr' (die bekend verondersteld worden), komen bij de GeoAssembler de volgende begrippen kijken: **labels, constanten, getallen, operatoren,** (deze vier vormen gecombineerd **expressies**) en tot slot zg. **'directives'**. De term 'directives' is overigens volstrekt uitwisselbaar met de term 'commandos'. Ik zal al de genoemde begrippen bespreken, te beginnen met het volgende voorbeeld:

```
; "5 x 9 = 42" says Peter to
    the artvark
```

Zoals u waarschijnlijk al gevat heeft gaat het hier om commentaar. Na een puntkomma wordt de rest van de regel in GeoAssembly genegeerd. Op naar het volgende voorbeeld:

```
label1:     lda #10
            nop
            jmp label1
label2:     nop
            nop
```

Labels definiëren

Een label is een naam met daarachter een dubbele punt. Let wel op: de naam mag maar max. 8 tekens lang zijn! De rest van de naam wordt er als hij langer is gewoon afgehakt. Had ik 'labeltje1' en 'labeltje2' als voorbeeld genomen, dan zou GeoAssembler hierop een 'dubbele labeldefinitie'-fout gegeven ('labeltje1' en 'labeltje2' worden namelijk allebei 'labeltje'). Vaak is het zo dat je zoveel labels nodig hebt, dat het moeilijk wordt ze namen te geven. Bovendien betreffen dat vaak labels die heel erg gewone dingen beschrijven, bijvoorbeeld 'lus_1:', of 'hoogtellerop:'. Als je zo'n label later weer definieert, dan komt er natuurlijk weer een dubbele labeldefinitie-fout. Om dit ongemak op te lossen zijn de zogenaamde lokalelabels ontwikkeld, onbelangrijke labeltjes voor onbelangrijke dingen. En wat dies meer zij: de assembler is hun bestaan alweer vergeten na het tegenkomen van een echt label. Om dit even te illustreren:

```
label1:     lda #10
1$:         nop
            jmp label1
label2:     nop
1$:         nop
```

Lokale labels zijn altijd getallen en eindigen zoals u ziet altijd met $. Er is hier ook geen sprake van een dubbele definitie-fout, want de assembler is het bestaan van de eerste '1$:' al weer vergeten als hij bij de tweede komt. Een nadeel van dit vergeten is natuurlijk dat je alleen tussen de eerste twee omliggende labels gebruik kunt maken van de lokale labels. Tot slot is er nog een lastig puntje: normaal worden alle namen van labels ook uitgevoerd naar de andere files waaraan de programma-file later in het link-proces gekoppeld wordt. Dit kan vermeden worden met het commando '.noglb'. Vanaf het punt waar dit staat worden nieuw gedefinieerde labels niet meer aan de linker bekendgemaakt. Om weer op de situatie met wel label-uitvoer over te gaan is er het commando '.glb'. Standaard, dus zonder van deze twee commando's gebruik te maken staat GeoAssembler op de '.glb'-stand, dus worden alle labels naar de linker 'geëxporteerd'.

Constanten

Naast labels kan een programmeur ook constanten invoeren. Dit werkt vaak verhelderend voor iemand die uw assembly listing probeert te lezen (zoals uzelf). Constanten definiëren gaat zo:

```
waar = 1
fout = 0
```

Let wel op bij het gebruik van deze constanten. Als een constante een waarde voorstelt moet er altijd een hekje (#) voor, als een constante een adres voorstelt niet. Dus:

```
lda #waar   wordt: lda #1,
en
lda waar wordt: lda $01
```

Om het allemaal wat moeilijker te maken kunnen constanten op twee manieren gedefinieerd worden, ten eerste zoals hierboven gedemonstreerd is met een enkele '='. Het is ook mogelijk met een dubbele '=', dus '=='. Het verschil zit hem erin, of later de GeoDebugger de constante kent. Constanten die met '=' gedefinieerd zijn kent hij niet, diegenen die met '==' gedefinieerd zijn wel. Maar hoe zit het dan met de GeoLinker?, vraagt u zich misschien af (waarschijnlijk echter niet). Standaard kent de GeoLinker alle gedefinieerde constanten, dit is uit te zetten met '.noeqin' en weer aan met '.eqin'. Dit alles net zoals het met '.glb' en '.noglb' (bij de labels) ging.

Opcodes en operanden

Met opcodes worden de termen 'lda', 'stx' e.d. bedoeld. De variabele die er vaak na komt heet de operand. Deze operand heeft vele verschijningsvormen, afhankelijk van het soort adressering. De GeoAssemler ondersteunt die natuurlijk allemaal. Ze hebben mooie engelse namen:

```
implied:                      (niets)
relative:                     adres
accumulator:                  a
absolute:
   zp-adress
absolute indexed X:
   zp-adress,x of adress,x
absolute indexed Y:
   zp-adress,y of adress,y
indexed indirecr:
   (zp-adress,x)
indirect indexed:
   (zp-adress),y
```

Operatoren en expressies

Getallen kunnen in assembly een waarde of een adres betekenen. In de GeoAssembler gelden dezelfde regels voor beide soorten getallen. Expressies zijn combinaties van getallen (of constanten of adreslabels, dat is allemaal hetzelfde) en operators. Een operator is bijvoorbeeld het '+' teken. Een expressie is dan dus bijv.: '$10 + 3'. Eerst definieer ik nu de getallen, er zijn 5 getal-modes:

1) decimal, beginnend met cijfer of #, bestaand uit cijfers, bijv. 91;
2) hexadecimal, beginnend met $, bestaand uit cijfers of a,b,c,d,e of f, bijvoorbeeld $1a (=decimaal 1*16+a=26, want a=10);
3) octal, beginnend met ?, bestaand uit cijfers 0-7, bijvoorbeeld ?76 (=decimaal 7*8+6=63);
4) binary, beginnend met %, bestaand uit nullen en enen, bijv. %1001000 (=decimaal 128+16=144);
5) character, beginnend en eindigend met ', en bestaand uit karakters, bijv. 'a'.

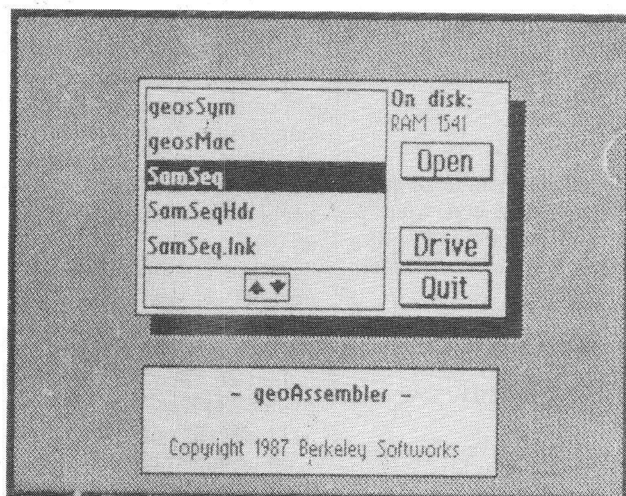
In illustratie 1 ziet u een overzicht van alle operators die in berekeningen gebruikt kunnen worden. Het veld 'precedence' geeft aan wie voorrang heeft op wie, waarbij het laagste getal steeds voorrang heeft. In berekeningen kunt u echter alles waarvan de beschrijving begint met 'logical' niet gebruiken. Dit zijn operators voor 'logical' expressions. Even nog wat voorbeelden van gewone expressies:

```
[4*(screen + $500 + %1001001)
    | 255
100+lus_1 / $16
#$01 //¯6
```

Logische expressies

In logical expressions, slordig ook wel 'Boolean' expressions genoemd, draait alles om 'ware' of 'onware' beweringen. Er zijn dus 2 waarden: 'waar' of 'onwaar', oftewel 1 of 0. Deze elementen zullen een ieder onder u die ook in PASCAL, MODULA of C programmeert, bekend in de oren klinken. Maar wat heeft dit nu met assembleertaal te maken? Dergelijke boolean-expressies komen toch alleen aan de orde als er ook if..then beweringen zijn, en die heeft machinetaal niet. Dat is ook wel zo, maar toch ook weer niet. de GeoAssembler heeft namelijk de '.if', '.elif' en '.endif' commando's. Het gaat bij deze keuzecommando's echter NIET om beslissingen die genomen moeten worden tijdens dat het programma DRAAIT, maar om beslissingen terwijl het programma VERTAALD wordt. Het gaat eigenlijk ook maar om 1 soort beslissing: behoort het stukje code na deze if tot het programma of niet. Om een voorbeeld te geven:

```
.if (1=0)
 lda #10
.elif (2=0)
 lda #20
 .else
  lda #34
 .endif
```



Conditionele code

Aangezien 1 niet gelijk is aan 0 of 2 zal de GeoAssembler in het bovenstaande stukje code 'lda #34' vertalen. De code die GeoAssembler ziet is dus afhankelijk van de waarde van de expressie bij de 'if'-s. Daarom spreekt men wel van 'conditional code' oftewel conditionele code. Het algemene principe is dus dat er eerst een '.if'-commando komt met daarachter op dezelfde regel een logische expressie. De regels daarna tot er een '.endif' of een '.elif' (else if) of een '.else' komt, zijn de conditionele code. Een ander voorbeeld, met de eerder genoemde variabele Pass1:

```
.if Pass1
.include GeosSym
.endif
```

Voor wie het nog niet weten mocht: het commando '.include filenaam' voegt de file 'filenaam' op de plaats van het '.include'-commando in de file in (wat 'filenaam' ook moge bevatten). Het bovenstaande voorbeeld heeft tot gevolg dat de file GeosSym alleen bij de eerste pass geinclude wordt, immers: alleen dan is Pass1 waar. De standaard-file GeosSym bevat alleen constanten en het weglaten van deze file betekent een grote tijdsbesparing bij het assembleren. Maar dat terzijde.

Sector-directives

De GeoAssembler kent 3 soorten geheugensectoren:
1) psect - voor de programmacode;
2) ramsect - hier kunnen datalabels gedeclareerd worden;
3) zsect - ook voor data, maar dan alleen in de zero-page ($0000- $00ff).
Deze scheiding in 3 gebieden wordt consequent doorgevoerd, pas tijdens het linkproces worden de preciese geheugenplaatsen vastgelegd, en dit gebeurt voor psect en ramsect apart. Meer hierover in het volgende deel, wanneer het over de GeoLinker gaat. Waar de zsect ligt staat natuurlijk altijd vast, en deze hoeft dan ook niet tijdens het link-proces van een geheugenplaats voorzien te worden. Om in de drie gebieden te kunnen werken zijn er 3 commando's: '.psect', '.ramsect', en '.zsect'. De code na een van deze commando's wordt dan opgevat als resp. psect, ramsect of zsect-code. Na '.psect' komt gewoon het programma. De directives '.ramsect' en '.zsect' zijn echter speciaal.

Ramsect

Deze sector is het datagebied, waar de programmeur variabelen en andere data op kan slaan. Na '.ramsect' kan de gebruiker dan beginnen met data-gebieden in gebruik te nemen:

```
.ramsect
variabel1:      .block  #3
variabel3:      .byte   #255,#1
variabel4:      .word   #255,#1
```

De namen 'variabel1', 'variabel1' en 'variabel2' kunnen nu in het psect deel gebruikt worden. Variabel1 wijst naar het begin van de ramsect, variabel3 naar het begin+3, en variabel4 naar het begin+5. Voorts zijn er 3 commando's gebruikt. In de eerste plaats is dat '.block x', dit commando doet helemaal niets, het neemt alleen x bytes in beslag, in het bovenstaande geval dus 3. Het '.byte x' commando slaat de meegegeven getallen als een byte achter elkaar op. Het commando '.word x' slaat de meegegeven waarden als words op, dus als getallen van 2 bytes, waarbij het eerste getal de low-byte is , en het tweede de high-byte is.
Het uiteindelijke getal wordt dus verkregen door low-byte+high-byte*256 te doen. Het voorbeeldje zou het volgende tot resultaat gehad hebben:

```
variabel1 = ramsectbegin
variabel3 = ramsectbegin+3
variabel4 = ramsectbegin+5
en geheugenplaats, waarde:
     $ramsectbegin+0, ??
     $ramsectbegin+1, ??
     $ramsectbegin+2, ??
     $ramsectbegin+3, 255
     $ramsectbegin+4, 1
     $ramsectbegin+5, 255
     $ramsectbegin+6, 0
     $ramsectbegin+7, 1
     $ramsectbegin+8, 0
(?? betekent: onbekend)
```

Overigens is het ook mogelijk om karakters achter '.byte' en '.word' te plaatsen, dit gaat bijvoorbeeld zo:
.byte "hallo allemaal",0
en heeft tot resultaat dat de tekst 'hallo allemaal' byte voor byte in het geheugen gezet wordt, afgesloten met een 0.

Zsect

Zsect werkt eigenlijk precies hetzelfde als '.ramsect', met één verschil: het heeft alleen betrekking op data gelegen in de zero- page. Omdat de plaats al vastligt (zero-page) is .zsect niet relocatable, en hoeft het niet bij het linken ergens in het geheugen vastgepind te worden. De offset in de zeropage wordt daarom direct achter het commando gezet, dus: '.zsect x'. Dan is x (een getal tussen 0 en 255) het begin van de zsect. Voor de rest werkt alles hetzelfde als met '.ramsect'.

Tot slot

Dit artikel begint nu al behoorlijk volumineus te worden, en het is dus tijd om te stoppen. Een overzichtje van alle besproken GEOS directives staat in illustratie 2. Illustratie 3 bevat een tabel die ik eigenlijk in het tweede deel van deze cursus al had willen plaatsen, namelijk een overzicht van alle GEOS-routines. In de volgende aflevering, dat is alweer de tiende, zal ik verdergaan op dit onderwerp. Dan komt namelijk het staartje van de GeoAssembler aan bod, met daarin o.a. de vorige keer al beloofde macro's en headers. Tevens zullen dan de andere delen van het GeoProgrammer-pakket, namelijk de GeoLinker en GeoDebugger aan de orde komen. Ook is er dan aandacht voor de speciale GEOS-files GeosSym en GeosMac. Tot de volgende keer dus!

Peter Boncz

GEOS-INFO

Aangezien Bert Venema voor langere tijd het ruime sop (letterlijk, hij werkt bij de marine, red.) heeft gekozen, neem ik de GEOS INFO-rubriek voor een tijdje van hem over. Er was deze maand een behoorlijk aantal reacties van lezers op allerlei actuele GEOS-zaken. Beginnen we bij Tim Jansma, hij vuurt werkelijk een indrukwekkende serie vragen op mij af.

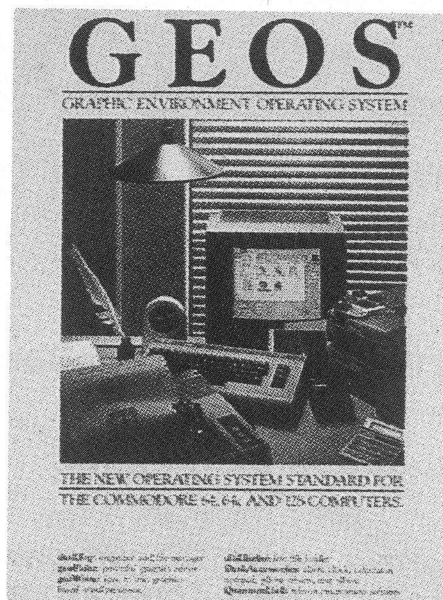
Het is inderdaad waar dat de applicaties Deskpack Plus, Fontpack Plus, Geocalc, Geochart, Geopublish en GeoProgrammer ook voor versie 1.3 werken. Sterker nog: ze werken ook voor versie 1.0, 1.2 en 2.0! Op de vraag of Geos versie 1.3 ergens omgeruild kan worden in versie 2.0 is het antwoord : ja, dit zal waarschijnlijk op de volgende CI-beurs weer mogelijk zijn. Je kan het natuurlijk ook proberen bij de firma die je Geos 1.3 verkocht heeft. Dit laatstgenoemde adres is waarschijnlijk ook de beste plaats om heen te gaan om je kapotte bootdisk weer gerepareerd te krijgen.

PageFox

Tenslotte vraagt Tim ons nog waar het 'DTP'-pakket PageFox te krijgen is. De GEOS-rubriek is niet echt het podium voor zulke vragen, maar goed, mijn advies is om eens rond te kijken bij bijv. Cat en Korsh, hier voorin het blad.

RAM 1764

C. Varkevisser uit Sassenheim stuurde ons 2 pagina's op met tips, trucs, wetenswaardigheden, filosofieën, etc.. In de eerste plaats vertelt hij ons dat de foto's bij het artikel over de RAM 1764 niet de genoemde REU voorstelden, maar een Commodore 128 met insteekmodem! Dat is inderdaad een vergipsinkje van de CI. (Het genoemde artikel was overigens niet van Bert of mij, en had slechts zeer zijdelings met GEOS te maken.) Zoals bekend (of misschien ook niet) is het erg moeilijk om in Nederland aan de commodore geheugenuitbreidingen (REU's), zoals de 1764, te komen.
C. Varkevisser is echter voor geen kleintje vervaart en heeft daarom zijn oplossing in het buitenland gezocht. Wat dus



gedaan? Een duits computerblad gekocht en in de advertentiepagina's gekeken. Een firma, genaamd DATA 2000 bleek bereid te zijn het produkt (een REU dus) te leveren voor het vriendelijke prijsje van 100 DM. Na het opsturen van een cheque met dit bedrag en het achteraf betalen van *f* 21,50 aan invoerrechten bleef er toch een aardige deal over.
Zeker omdat de verkoper van V&D, die dhr. Varkevisser eerst geconsulteerd had, weliswaar gemeld had dat het produkt niet voorradig was, maar dat er toch ongeveer zo'n *f*280,- voor neergeteld zou moeten worden. Al met al weer een bewijs dat het erg nuttig kan zijn om eens over de grens te kijken, zeker omdat C. VarkeVisser ons voorrekent dat het gemiddelde GEOS-pakket bij onze oosterburen stukken goedkoper is dan hier of in Engeland. Dit soort zaken is echter niet alleen rozengeur en maneschijn, er kunnen van allerlei dingen misgaan (het opsturen van een cheque), zeker als je de duitse taal niet goed beheerst.

DeskTop

Cools Patrick uit België meldt ons 'dat hij het toetsenbord van zijn 64 wel wat warmgehouden heeft'. Dit resulteerde onder andere in de tip om het DeskTop-programma op de datadisk te zetten. Hij gaat ervan uit dat u met 2 drives werkt. Dus op de bootdisk hoeft volgens hem geen desktop te staan. Hier heeft hij gelijk in: als GEOS de DeskTop op drive 8 niet vindt, en ook drive 9 is in gebruik, dan wordt eerst hier nog gezocht alvorens een error te rapporteren. Cools (ik hoop werkelijk dat ik het goed begrepen heb, en ik je naam nu niet omdraai) stipt bovendien nog een heikel puntje aan; een oude koe uit de GEOS-sloot.

> Heeft U een vraag, idee of een reactie? (betreffende GEOS, dit is niet lieve Lita)
> stuur hem naar:
> GEOS-INFO rubriek
> Postbus 43048
> 1009 ZA Amsterdam

CTRL-H

Het gaat om het hardcopy-programma, u weet wel, uit de vierde en vijfde aflevering van de CI. De CTRL-H routine die na het indrukken van CTRL-H het 'hardcopy.prg' programma moet starten werkt volgens hem niet. Misschien ligt dit aan het feit dat hij het verkeerde deel van de listing gebruikt, deze verschilde voor versie 1.x en versie 2.0. Een andere, waarschijnlijkere, oorzaak is een klein foutje in de listing van hardcopyLst (uit de vierde aflevering). Na '.byte "hardcopy.prg" ' had nog ',0' moeten staan. Als er dan voor gezorgd wordt dat er inderdaad een programma, met de naam 'hardcopy.prg' (de listing uit de vijfde aflevering) op de disk staat, (op het moment dat er op CTRL-H gedrukt wordt), dan moet het werken.

GEOS PD 2

In het vorige nummer van de CI is de aankondiging van de GEOS PD Disk 2 een beetje de mist ingegegaan. We proberen het opnieuw:

Bestel nu de GEOS PD Disk 2 door *f* 12,- over te maken op girorekening: 3157656, t.n.v. Infolist Huizen onder vermelding van: 'GEOS PD 2'

Het netelige feit zit hem echter hierin dat in sommige gevallen de CTRL-H routine inderdaad problemen kan veroorzaken. GEOS staat een dergelijke constructie namelijk niet toe. Er is geen mogelijkheid om een programaatje ergens in het geheugen te houden dat continu kijkt of er een toets ingedrukt is. GEOS kan maar met 1 programma tegelijk werken en kent geen geheugenbescherming. En dat is nu juist wat we nodig hadden. De CTRL-H routine maakt gebruik van een stuk geheugen dat alleen in gebruik is als er met bijzondere drives gewerkt wordt. Als hiermee gewerkt wordt, dan is er echter een goede kans (zo'n 100%) dat de zaak in de soep loopt. Een oplossing is dan het 'hardcopy.prg'-programma om te werken tot een DESK_ACC, maar dan kunnen er alleen hardcopies gemaakt worden op momenten dat een DESK_ACC gestart kan worden, dus niet op ieder moment en niet in iedere applicatie. En dat zijn nu juist eigenschappen die voor een schermdump-programma van belang zijn.

1541-dump

Cools geeft ons verder een wijziging in de listing van de QuickTop (nu echt een hele tijd terug), die dit programma ook geschikt maakt voor 1.3 en 2.0. Een soortgelijke wijziging heb ik al een paar maanden terug doorgevoerd, en de QuickTop die nu op de PD-schijf staat, werkt met alle versies van GEOS. Als illustratie plaatsen we nog de listing van Cools Patrick voor een dump-programma. Dit is zijn bijdrage aan het besproken hardcopy-probleem. Het behelst een DESK ACCESSORY genaamd 1541-dump, die bij aanroep een file 'dump' met daarin een hardcopy van het beeldscherm naar disk schrijft. Met deze file kan de gebruiker dan later doen wat hij of zijzelf wil (op een of andere manier uitprinten, naar GeoPaint halen, etc). Maar let op: ook deze 'hardcopy mogelijkheid' werkt alleen maar op momenten dat een desk accessory gestart kan worden!

Peter Boncz

Basic miniatuurtjes

Een rubriek van Alex van Maarschalkerwaart

Gevraagd

Basicminiatuurtjes

Stuur al uw kleine, leuke, nuttige (Amiga en C64) programmatjes naar:
Sala Communications, Postbus 43048, 1009 ZA Amsterdam o.v.v. MINIATUURTJES.

Datum berekenen.

Dit programma berekent de dag waarop een gegeven datum valt (bijvoorbeeld geboortedag). Wel moet de datum tussen 1900 en 2000 liggen.

```
1   rem--datum berekenen--
2   rem--sven verstrepen--
5   printchr$(147)
10  print"tik[SPACE]datum[SPACE]in[SPA
    CE](dd,mm,jj)"
20  inputd,m,j
30  j=j-1900
40  if m=1then70
50  if m=2then70
60  goto100
70  j=j-1:m=m+13:goto90
80  m=m+1
90  k=int(365.25*j)
100 l=int(30.6*m)
110 z=k+l+d+72926
120 z=z/7-int(z/7)
130 ifz=0then160
140 z=int(z*7+.01)
150 print"[SHIFT-CLR]"
160 print"deze[SPACE]datum[SPACE]valt[
    SPACE]op[SPACE]een";
170 ifz=0thenprint"[SPACE]dinsdag":goto240
180 ifz=1thenprint"[SPACE]woensdag":goto240
190 ifz=2thenprint"[SPACE]donderdag":goto240
200 ifz=3thenprint"[SPACE]vrijdag":goto240
210 ifz=4thenprint"[SPACE]zaterdag":goto240
220 ifz=5thenprint"[SPACE]zondag":goto240
230 ifz=6thenprint"[SPACE]maandag":goto240
240 print"***************************
    *************"
250 print"[SPACE]druk[SPACE]space[SPAC
    E]voor[SPACE]herbeginnen":goto260
260 geta$:ifa$=""then260
270 ifa$="[SPACE]"then5
```

Programma beveiliging.

J.D. Brugge uit België, heeft dit programma gemaakt om al zijn ideeën te beveiligen.
Het creëert een programma op disk (voor tape wordt regel 10 open1,1,1"compile!"), "COMPILE", dat het belangrijkste beveiligingswerk doet.
Maak en save een programma,--In basic moet voor het eind v/h prg (end,new,load"andere prog",...)poke 43,1 staan.
Load "COMPILE",8,1--Poke 43,34--Load "prog.naam",8 (niet,8,1!!)--Poke43,1---Save"Anderenaam",8---Run om te testen. Als alles goed is, moet het programma nu normaal werken, met als enige verschil dat:Run/Stop(Restore) niet meer werken en bij List het volgende staat: 1989 sys 2006 Text.

```
0   rem * (c)1989 the demo master/corp
    .   against bad crackings
1   rem ---j.d.brugge---
10  open 1,8,1,"compile!,p,w"
20  fork=1to33:reada:print#1,chr$(a);:
    t=t+a:next:close1
30  ift=2391thenprint"you[SPACE]can[SP
    ACE]try[SPACE]now,i'm";:end
40  print"?data[SPACE]error-check[SPAC
    E]listing";:list
50  open 1,8,15,"s:compile!":close1
60  data 1,8,15,8,197,7,158,50,48,54,54
70  data 32,84,69,89,84,,,,169,34,133,
    43,169,225
80  data 141,40,3,32,89,166,76,174,167
```

Directory.

Dit programma is enkel bedoeld als voorbeeld van hoe op een eenvoudige wijze de directory uit te lezen is.
Regel 10 opent de Dir als een bestand op disk (Dos interpeteert dit als load"$").
De [rev-on] in 20 zorgt voor 't reversed afdrukken van de header.
De volgende regels lezen de tekens en zetten de Filenaam en prg-soort op het scherm.

```
0   rem ---directory---
1   rem ---j.d.brugge--
2   input"[SHIFT-CLR]geef[SPACE]het[SP
    ACE]devicenummer";dn:if dn <8goto2
10  open 1,dn,0,"$"
20  print"[CTRL-2][SHIFT-CLR][CTRL-9]"
    ;:poke 53281,0
30  get#1,a$:ifa$<>chr$(34)anda$<>chr$
    (199)goto30
40  if a$=chr$(199)thenclose1:end
50  printa$;:fork=1to25:get#1,a$:print
    a$;:next:print:goto30
```

Schermgekte.

Dat kleine programmeerfouten ook leuke gevolgen kunnen hebben, wil Kees Patijn uit Rotterdam ons met dit miniatuur-programma duidelijk maken.
Je kan het met Sys 828 opstarten.

```
10  data120,238,32,208,123,33,208,166,
    203,224,64,208,244,76,60,3,0:print
    chr$(147)
20  fort=0to16:reada:poke828+t,a:next
```

UGA's Musical Enlightenment

Software voor de kleine beurs

United Graphic Artists, kortweg UGA, is al een tijdje bezig om goedkope, maar niet amateuristische software aan de man te brengen. Rob Fonteine van UGA is niet alleen in Nederland druk bezig, ook in het buitenland weten velen al van zijn bestaan. Doordat Rob zich veel in het buitenland beweegt, heeft hij zo zijn contacten opgebouwd. Vanuit die contacten brengt Rob Fonteine in samenwerking met een Brit, het nieuwsmagazine Newsflash op de markt, maar ook goedkope software van professionele alure.



Het net nieuwe programma Music Enlightenment maakt deel uit van een serie Amiga programma's, die zich qua mogelijkheden kunnen meten met de vaak duurdere software van andere producenten. UGA wil door deze programma's ook veel nieuw talent een kans geven, die bij de grotere producenten geen kans krijgen.

Nederlands produkt

Musical Enlightenment werd geschreven door de Nederlandse programmeur Niek Luuring en is voorzien van een Nederlandstalige handleiding. Het programma is verdeeld in vier secties. Met de song-module worden de noten ingevoerd en het gewenste instrument gekozen. Dit wordt gedaan in zogenoemde sporen. Voor iedere Amiga stem (totaal twee links en twee rechts) staat er één spoor ter beschikking, zodat ook andere instrumenten aan de hoofdmelodie kunnen worden toegevoegd. In deze sectie is het ook mogelijk de stemmen aan/uit te zetten en kan het audio-filter worden uitgeschakeld; daardoor zijn er meer hoge tonen hoorbaar. Dit is bij een aantal muziekdemo's en programma's ook zichtbaar als de power-LED uitgaat. Als extra mogelijkheid kan de snelheid van het afspelen in de song-module worden veranderd.

Instrumenten

De instrument-module is het gedeelte waar met de instrumenten geëxperimenteerd kan worden. Met de muis kunt de attack, decay, sustain en release veranderen, zodat het instrument precies zo klinkt als de bedoeling is. Deze functies zijn vergelijkbaar met die in Eagis Sonix. In deze sectie kan ook een effect worden verbonden met een instrument. Zo zijn bijvoorbeeld een watereffect of een echo mogelijk, dat aan één bepaald instrument kan worden gekoppeld.

Effecten

Met de effecten-module kunnen zelf effecten worden samengesteld met behulp van diverse golfvormen. De belangrijkste effecten zijn: vibrato en tremolo. Ook is het mogelijk op een toetsenbordje op het scherm een zogenoemde arpeggio te maken. Een arpeggio bestaat uit een aantal noten, die tegelijk worden aangeslagen, zodat een bredere klank hoorbaar is.

Sample

Tenslotte kan de gebruiker met de sample-module, ook eigen gesampelde muziekinstrumenten kan gebruiken. Voor hen die geen sampler bezitten, zijn er in de UGA Public Domain-serie en andere bekende series als Fish voldoende instrumenten te vinden die te gebruiken zijn om een muziekstuk te componeren.

In deze sectie kan men samples zodanig bewerken dat er nieuwe ontstaan. Bepaalde stukjes kunnen worden uitgesneden, gekopieerd, gemengd en kunnen er ook berekeningen op los worden gelaten. Een ingeladen soundsample wordt grafisch op het scherm getoond waarna er in/uit vergroot kan worden en op een makkelijke manier veranderd kan worden.

Als extras heeft UGA een afspeelroutine voor Devpac en Aztec als sourcefile toegevoegd, zodat programmeurs de gemaakte muziek in eigen programmatuur kunnen inbouwen. Voor de niet-programmeurs kan de zelfgemaakte muziek met een losse afspeelroutine hoorbaar worden gemaakt.

Tot slot

Al met al biedt dit pakket zeer veel mogelijkheden voor het maken van eigen computermuziek. UGA heeft al toegezegd, dat in een volgende versie een MIDI-sectie zal worden toevoegd, zodat er ook met synthesizers kan worden gewerkt. Het complete pakket kost *f* 59,-. Beslist niet duur en een aanrader voor Amiga muziek-fans.

Checksum C-64

Syntax Checksum

Het overtikken van een listing kan een heel karwei zijn en als u een beetje normaal mens bent dan maakt u daarin beslist een aantal fouten. Nu is niets moeilijker om de fouten uit je eigen werk te halen. Al geruime tijd heeft Jan Bodzinga hiervoor een zgn. Checksum-programma geschreven. Om de vele nieuwe lezers van Commodore-info te helpen volgt hieronder nog een keer een volledige uitleg over de werking van dit programma, waarmee het, hoe vreemd dat misschien ook lijkt, echt mogelijk is om met behulp van dit programma de fouten in elke door ons geplaatste listing op te sporen.

Hiervoor gaat u als volgt te werk:

1. U tikt de listing heel zorgvuldig over en SAVEt hem voordat u het programma RUNt op een diskette of cassette.
2. U tikt het RUN commando in. Mocht het programma de boodschap 'FOUT in dataregels!' geven dan heeft u een fout bij het overtikken gemaakt. Herstel dan de fout en SAVE de verbeterde versie. Mocht het programma met de boodschap 'data is weggezet checksum testen met sys...'komen dan is tot dusver alles goed. Het programma is nu in een stukje machinetaalgeheugen gezet. Als u het NEW commando geeft blijft het toch in de computer staan.

Alle door ons geplaatste programma's zijn in Basic geschreven.

Als u een programma heeft overgetikt SAVE het eerst, mocht er iets mis gaan dan hoeft u niet de gehele listing opnieuw te gaan intikken. Als u nu een programma op fouten wilt gaan controleren dan kunt u dat in het geheugen laden(wel eerst het checksum programma hebben gerund). Vervolgens typt u zonder het programma te runnen de opdracht sys 49152(c-64) of sys 1536 (c-16 en plus/4)in.

Als alles goed is gegaan loopt er nu een rij regelnummers over het scherm met getallen erachter. Dezelfde lijst staat ook achter elk door ons geplaatste programma. Wijkt nu een nummer achter een regelnummer af van het nummer dat in het blad staat dan heeft u in die regel iets anders ingetikt dan er in het blad stond. U kunt de stroom getallen d.m.v. de RUN/STOP toets pauzeren en weer vervolgen met de F1 of F7 toets. Het is uitermate belangrijk dat u goed met dit programma overweg kunt en mocht u het niet goed werkend krijgen bel dan gerust even met onze listingservice telefoonlijn. (Maandag 17.00 - 21.00 uur. Telefoonnummer 02155-25162.)

De laatste tijd wordt er weer veel gebeld zodat U nogwel eens in gesprek krijgt, daarom houdt uw vraag kort, vermeld in welk blad het desbetreffende artikel stond. Heeft U veel vragen, of zis uw vraag erg uitgebreid, doe het dan schriftelijk, zodat we veel mensen op de maandag avond te woord kunnen staan.

```
1   rem ********************************
2   rem basic loader "SYNTAX.CHECKSUM"
3   rem na de commando's "run" en "new"
4   rem blijft dit programma in het ge-
5   rem heugen. laad het te testen pro-
6   rem gramma en tik daarna sys 49152.
7   rem ********************************
10  i=49152  :rem beginadres
20  reada:ifa<0then40:rem data ingelezen
30  pokei,a:i=i+1:b=b+a:goto20
40  if b<>16844thenprint"[SHIFT-CLR]fo
    ut[SPACE]in[SPACE]dataregels!":b=0
    :end
50  poke49184,148:poke49185,192
55  i=49300
60  read a: ifa<0then80
70  pokei,a:b=a+b:i=i+1:goto60
80  if b<>20068thenprint"[SHIFT-CLR]fo
    ut[SPACE]in[SPACE]dataregels![SPAC
    E](vanaf[SPACE]regel[SPACE]240)":b
    =0:end
90  print"data[SPACE]is[SPACE]weggezet"
95  print"checksum[SPACE]testen[SPACE]
    met[SPACE]sys49152"
100 data 165,43,166,44,133,163,134,164
    ,169,147
110 data 32,210,255,160,0,240,3,32,73,192
120 data32,73,192,208,1,96,32,225,255,208
130 data 3,76,116,164,32,81,192,32,73,192
140 data 240,12,201,32,240,247,24,101,
    167,133
150 data 167,76,37,192,166,167,169,0,1
    32,168
160 data 32,205,189,169,13,32,210,255,
    164,168
170 data 76,17,192,200,208,2,230,164,1
    77,163
180 data96,162,0,189,123,192,240,6,32,210
190 data 255,232,208,245,32,73,192,170
    ,32,73
200 data 192,132,168,32,205,189,162,3,
    169,32
210 data 32,210,255,202,208,250,169,0,
    133,167
220 data 164,168,96,82,69,71,69,76,32,0
230 data  -1
240 data 165,197,201,3,240,7,201,4,240
250 data 6,76,148,192,76,34,192,169
260 data 147,32,210,255,76,161,192
270 data -1
```

** EINDE LISTING checksum 64 **

```
REGEL   1   249          REGEL 100   183
REGEL   2   84           REGEL 110   158
REGEL   3   105          REGEL 120   232
REGEL   4   2            REGEL 130   183
REGEL   5   246          REGEL 140   96
REGEL   6   152          REGEL 150   96
REGEL   7   249          REGEL 160   127
REGEL  10   157          REGEL 170   71
REGEL  20   64           REGEL 180   223
REGEL  30   38           REGEL 190   73
REGEL  40   57           REGEL 200   79
REGEL  50   14           REGEL 210   109
REGEL  55   251          REGEL 220   106
REGEL  60   192          REGEL 230   225
REGEL  70   42           REGEL 240   16
REGEL  80   244          REGEL 250   163
REGEL  90   245          REGEL 260   92
REGEL  95   237          REGEL 270   22
```

PRINT OUT C-64 met o.a. Rekenwonder

LINGO 64

Ditmaal weer een lange listing. Het zal de nodige uurtjes kosten om dit programma van S.F. de Vries uit Oosterwolde over te tikken. U zult zien het is zeker de moeite waard. Is het U te veel werk of durft U het niet aan, bestel dan de Infolist diskette waar o.a. dit programma opstaat. Ook op deze diskette staat een Seq. file met een groot aantal woorden, die hoeft U dan niet meer in te voeren.

Het spel wordt gespeeld met twee spelers. U kunt een groot aantal standaard woorden invoeren (maximaal 100). Deze worden gesaved op diskette dus dit hoeft maar 1 keer te gebeuren. Na het opstarten van het spel, wordt er om de namen van de spelers gevraagd. Na het invullen wordt een matrix op het scherm getekend door het indrukken van de spatiebalk. De eerste letter van het willekeurig gekozen woord wordt linksboven in de matrix geplaatst. De bedenktijd is beperkt, snelheid is dus geboden. (ongeveer 15 sec.) Wacht je te lang dan gaat de beurt over naar je tegenstander. Wacht deze ook te lang dan wordt het woord door de computer gegeven en gaat het spel met een nieuw woord verder. Letters die wel in het woord voorkomen maar niet op de juiste plek staan worden in het geel weergegeven. Letters die wel op de goede plaats staan zijn rood. Is het woord goed dan komt het lingo scherm weer te voorschijn en mogen er één of meerdere blokjes worden gepakt. Elke speler heeft 20 blokken, 17 met een getal en 3 rode. Wordt het rode blok gepakt dan gaat de beurt naar de tegenstander. Je hebt LINGO als er 5 blokken op een rij staan, horizontaal, vertikaal of diagonaal.

```
 5   rem woordspel lingo
10   rem door s.f. de vries
15   rem        oosterwolde
20   rem         voor c-64
25   poke646,0:printchr$(142)
30   l$="lingo":p1=0:p2=0:as=0:vc=54272
35   sp=int(rnd(0)*2)+1
40   rem a=a+1:ifa=1thenload"mach.lingo
     ",8,1
45   dim w$(100)
50   gosub1840
55   ne$="10121416182022242628303234363
     840424446485052545658"
60   no$="11131517192123252729313335373
     941434547495153555759"
65   gosub1960
70   fort=0to500:next:poke53281,15:poke
     53280,15
75   fort=0to250:next:poke53281,12:poke
     53280,12
80   fort=0to250:next:poke53281,11:poke
     53280,11
85   fort=0to250:next:poke53281,0:poke5
     3280,0
90   gosub2085
95   gosub1605
100  x=0:y=16:gosub1165:input"speler[SP
     ACE]1:";sp$(1)
105  y=19:gosub1165:input"speler[SPACE]
     2:";sp$(2)
110  as=as+1:gosub1085
115  k$=ne$:k=25:xx=3:yy=8:gosub740
120  k$=no$:k=25:xx=23:yy=8:gosub740
125  k$=ne$:k=25:pe=1347:gosub795
130  gosub1260:ifx=5thengoto110
135  ke$=k$+"000000":ke=20
140  k$=no$:k=25:pe=1367:gosub795
145  gosub1260:ifx=5thengoto110
150  ko$=k$+"000000":ko=20
155  sys49152
160  x=1:y=20:gosub1165:print"[CTRL-2]d
     ruk[SPACE]op[SPACE]spatiebalk[CTRL
      1]"
165  geta$:ifa$<>"[SPACE]"then165
170  gosub1180:sys49182
175  rem woord kiezen
180  x=1:y=8:gosub1165:printp1;"punten"
185  x=26:gosub1165:printp2;"punten"
190  lt=0:cc=0:aw$="":poke650,64
195  wk=int(rnd(0)*wc)+1
200  wk$=w$(wk)
205  rem
210  x=15:y=4:gosub1165
215  ifsp=1thenprint"[CTRL-8]<-----<<<[
     CTRL-1]"
220  ifsp=2thenprint"[CTRL-8]>>>----->[
     CTRL-1]"
225  iflt>0then245
230  rem eerste letter plaatsen
235  x=15:y=8:gosub1165
240  printleft$(wk$,1)
245  x=1:y=20:gosub1165:poke646,sp+5
250  print"[40xSPACE]"
255  x=1:y=20:gosub1165:print"welk[SPAC
     E]woord[SPACE]is[SPACE]het[SPACE]"
     sp$(sp)"[SPACE]:[SPACE][5xCOM-D][5
     xCRSR-LEFT]";
260  c=0:poke646,0
265  forn=1to5
270  poke649,1
275  c=c+1
280  geta$:ifa$=""ora$="[SPACE]"ora$=ch
     r$(20)then290
285  printa$;:aw$=aw$+a$:c=0:goto300
290  ifc>300thenc=0:goto305
295  goto275
300  nextn:poke649,0:forn=0to550:nextn:
     c=0:goto325
305  cc=cc+1:ifcc=2then goto505
310  ifsp=1thensp=2:goto320
315  ifsp=2thensp=1
320  aw$="":goto210
325  x=0:y=20:gosub1165:print"[SPACE]ge
     el[SPACE]is[SPACE]letter[SPACE]goe
     d[SPACE]-[SPACE]rood[SPACE]plaats[
     SPACE]goed"
330  rem woord in matrix zetten
335  x=13:y=8+(2*lt):gosub1165
340  fort=1to5
345  x=x+2:gosub1165
350  printmid$(aw$,t,1)
355  nextt:print"[CTRL-1]"
360  ifcc=2thengoto625
```

```
365  rem vergelijken keuzewoord en inge
     voerd woord
370  x=13
375  cw$=wk$:ca$=aw$
380  forn=1to5
385  x=x+2:gosub1165
390  ifmid$(cw$,n,1)<>mid$(aw$,n,1)then
     410
395  gosub1165:print"[CTRL-3]";mid$(aw$
     ,n,1);"[CTRL-1]"
400  ca$=left$(ca$,n-1)+"`"+right$(ca$,
     5-n)
405  cw$=left$(cw$,n-1)+"*"+right$(cw$,
     5-n)
410  nextn
415  ifaw$=wk$thenaw$="":goto520
420  x=13
425  fort=1to5
430  x=x+2
435  forc=1to5
440  ifmid$(cw$,c,1)<>mid$(ca$,t,1)then
     460
445  gosub1165:print"[CTRL-8]";mid$(ca$
     ,t,1);"[CTRL-1]"
450  ca$=left$(ca$,t-1)+"`"+right$(ca$,
     5-t)
455  cw$=left$(cw$,c-1)+"*"+right$(cw$,
     5-c)
460  nextc
465  nextt:forn=1to200:nextn
470  ifaw$=wk$thenaw$="":goto520
475  ifcc=1andlt=4thencc=cc+1
480  ifcc=2andaw$<>wk$then lt=4:goto495
485  aw$="":lt=lt+1:iflt<5then205
490  lt=4:cc=cc+1
495  ifsp=1thensp=2:goto505
500  ifsp=2thensp=1
505  ifcc=2thenaw$=wk$:print"[CTRL-7]":
     forc=0to1000:nextc:goto335
510  goto210
515  rem woord is goed blokjes pakken
520  ba=5-lt
525  fort=1to15:poke53280,t:forn=1to25:
     next:next
530  x=1:y=20:gosub1165
535  poke649,10
540  print"[CTRL-2][2xSPACE]druk[SPACE]
     spatie[SPACE]om"ba"[CRSR-LEFT][SPA
     CE]blokje(s)[SPACE]te[SPACE]pakken
     ![CTRL-1][SPACE]"
545  geta$:ifa$<>"[SPACE]"then545
550  sys49212
555  x=16:y=4:gosub1165
560  ifsp=1thenprint"[CTRL-8]<----<<<[C
     TRL 1]"
565  ifsp=2thenprint"[CTRL-8]>>>---->[C
     TRL 1]"
570  forb=1toba
575  poke646,b+3:poke649,1
580  x=0:y=20:gosub1165:print"[SPACE]dr
     uk[SPACE]spatie[SPACE]om[SPACE]"b"
     [CRSR-LEFT].e[SPACE]blokje[SPACE]t
     e[SPACE]pakken"
585  geta$:ifa$<>"[SPACE]"then585
590  poke646,0:poke649,0
595  ifsp=1thengosub870
600  ifsp=2thengosub980
605  ifk1$="00"thengoto620
610  ifx=5then660
615  nextb:goto625
620  ifk1$="00"thensp=sp+1:ifsp>2thensp
     =1:k1$="**"
625  x=0:y=20:gosub1165:print"[CTRL-2][
     SPACE]druk[SPACE]spatiebalk[SPACE]
     voor[SPACE]letter-matrix[4xSPACE][
     CTRL-1]"
630  poke649,1
635  geta$:ifa$<>"[SPACE]"then635
640  ifcc<>0thengoto655
645  x=19:y=12:gosub1165:print"[2xSPACE
     ]"
650  sys49152
655  sys49244:goto175
660  rem lingo
665  poke649,1
670  x=1:y=20:gosub1165:print"[CTRL-2]d
     ruk[SPACE]spatiebalk[SPACE]voor[SP
     ACE]de[SPACE]punten[7xSPACE][CTRL-
     1]"
675  geta$:ifa$<>"[SPACE]"then675
680  ifsp=1thenp1=p1+50
685  ifsp=2thenp2=p2+50
690  print"[SHIFT-CLR]":print"[CRSR-DOW
     N]puntenverdeling[SPACE]na[SPACE]"
     as"spelletjes:"
695  x=0:y=5:gosub1165:printsp$(1):x=15
     :gosub1165:printp1
700  x=0:y=10:gosub1165:printsp$(2):x=1
     5:gosub1165:printp2
705  x=0:y=15:gosub1165
710  print"verder[SPACE]spelen[SPACE]of
     [SPACE]stoppen?"
715  geta$:ifa$<>"v"anda$<>"s"then715
720  ifa$="v"thengoto110
725  print"[4xCRSR-DOWN]u[SPACE]bent[SP
     ACE]gestopt!!!!!"
730  poke650,0:poke649,10
735  end
740  rem matrix invullen
745  x=xx:y=yy
750  k1=2*(int(k*rnd(0))+1)
755  k1$=mid$(k$,k1-1,2)
760  poke211,x:poke214,y:sys58640
765  printk1$
770  if2*k=k1thenk$=left$(k$,k1-2):goto
     780
775  k$=left$(k$,k1-2)+right$(k$,2*k-k1
     )
780  x=x+3:ifx=xx+15thenx=xx:y=y+2
785  k=k-1:ifk=0thenreturn
790  goto750
795  rem kiezen door te halen nummers
800  a=0:p=pe
805  k1=2*int(k*rnd(0)+1)
810  k1$=mid$(k$,k1-1,2)
815  if2*k=k1thenk$=left$(k$,k1-2):goto
     825
820  k$=left$(k$,k1-2)+right$(k$,2*k-k1
     )
825  x=19:y=12:gosub1165:print"[COM-8]"
     k1$"[CTRL-1]"
830  fort=0to4
835  forn=ptop+12step3
840  ifpeek(1523)=peek(n)andpeek(1524)=
     peek(n+1)then poken,244:poken+1,97
845  nextn
```

```
850 p=p+80:nextt
855 p=pe:k=k-1:a=a+1:ifa=8then865
860 goto805
865 gosub1165:print"[2xSPACE]":return
870 rem kiezen blokken even
875 k$=ke$:p=1347
880 k1=2*int(ke*rnd(0)+1)
885 k1$=mid$(k$,k1-1,2)
890 if2*ke=k1thenk$=left$(k$,k1-2):got
    o900
895 k$=left$(k$,k1-2)+right$(k$,2*ke-k
    1)
900 ke$=k$:p=1347
905 x=19:y=12:gosub1165
910 ifk1$="00"thenprint"[CTRL-3][CTRL-
    9][COM-J][CTRL-0][COM-K][CTRL-1]":
    goto920
915 print"[COM-8]"k1$"[CTRL-1]"
920 fort=0to4
925 forn=ptop+12step3
930 ifpeek(1523)=peek(n)andpeek(1524)=
    peek(n+1)then 940
935 goto955
940 poken+vc,7:poken+vc+1,7
945 forc=0to1500:next
950 poken,244:poken+1,97
955 nextn
960 p=p+80:nextt
965 p=1347:gosub1260
970 p=1347:ke=ke-1
975 ke$=k$:return
980 rem kiezen blokken oneven
985 k$=ko$:p=1367
990 k1=2*int(ko*rnd(0)+1)
995 k1$=mid$(k$,k1-1,2)
1000 if2*ko=k1thenk$=left$(k$,k1-2):got
     o1010
1005 k$=left$(k$,k1-2)+right$(k$,2*ko-k
     1)
1010 x=19:y=12:gosub1165
1015 ifk1$="00"thenprint"[CTRL-3][CTRL-
     9][COM-G][CTRL-0][COM-K][CTRL-1]":
     goto1025
1020 print"[COM-8]"k1$"[CTRL-1]"
1025 fort=0to4
1030 forn=ptop+12step3
1035 ifpeek(1523)=peek(n)andpeek(1524)=
     peek(n+1)then1045
1040 goto1060
1045 poken+vc,7:poken+vc+1,7
1050 forc=0to1500:next
1055 poken,244:poken+1,97
1060 nextn
1065 p=p+80:nextt
1070 p=1367:gosub1260
1075 p=1367:ko=ko-1
1080 ko$=k$:return
1085 rem opzetten getallenmatrix
1090 print"[SHIFT-CLR][CRSR-DOWN][14xSP
     ACE][CTRL-9][SPACE]l[SPACE]i[SPACE
     ]n[SPACE]g[SPACE]o[SPACE][2xCRSR-D
     OWN]"
1095 x=2:y=4:gosub1165:printsp$(1)
1100 x=38-len(sp$(2)):gosub1165:printsp
     $(2)
1105 print"[2xCRSR-DOWN][2xSPACE][COM-A
     ]CC[COM-R]CC[COM-R]CC[COM-R]CC[COM
     R]CC[COM-S][4xSPACE][COM-A]CC[COM
     R]CC[COM-R]CC[COM-R]CC[COM-R]CC[C
     OM S]
1110 print"[2xSPACE]B[2xSHIFT-SPACE]B[2
     xSHIFT-SPACE]B[2xSHIFT-SPACE]B[2xS
     HIFT SPACE]B[2xSHIFT-SPACE]B[4xSPA
     CE]B[2xSHIFT-SPACE]B[2xSHIFT-SPACE
     ]B[2xSHIFT-SPACE]B[2xSHIFT-SPACE]B
     [2xSHIFT-SPACE]B
1115 print"[2xSPACE][COM-Q]CC[SHIFT-+]C
     C[SHIFT-+]CC[SHIFT-+]CC[SHIFT-+]CC
     [COM-W][4xSPACE][COM-Q]CC[SHIFT-+]
     CC[SHIFT-+]CC[SHIFT-+]CC[SHIFT-+]C
     C[COM-W]
1120 print"[2xSPACE]B[2xSPACE]B[SPACE][
     SHIFT-SPACE]B[2xSHIFT-SPACE]B[2xSH
     IFT SPACE]B[2xSHIFT-SPACE]B[SHIFT-
     SPACE][SPACE][2xSHIFT-SPACE]B[2xSH
     IFT SPACE]B[2xSHIFT-SPACE]B[2xSHIF
     T SPACE]B[2xSHIFT-SPACE]B[2xSHIFT-
     SPACE]B
1125 print"[2xSPACE][COM-Q]CC[SHIFT-+]C
     C[SHIFT-+]CC[SHIFT-+]CC[SHIFT-+]CC
     [COM-W][4xSPACE][COM-Q]CC[SHIFT-+]
     CC[SHIFT-+]CC[SHIFT-+]CC[SHIFT-+]C
     C[COM-W]
1130 print"[2xSPACE]B[2xSHIFT-SPACE]B[S
     HIFT SPACE][SPACE]B[2xSHIFT-SPACE]
     B[SHIFT-SPACE][SPACE]B[2xSHIFT-SPA
     CE]B[4xSPACE]B[2xSHIFT-SPACE]B[2xS
     HIFT SPACE]B[2xSHIFT-SPACE]B[2xSHI
     FT SPACE]B[2xSHIFT-SPACE]B
1135 print"[2xSPACE][COM-Q]CC[SHIFT-+]C
     C[SHIFT-+]CC[SHIFT-+]CC[SHIFT-+]CC
     [COM-W][SHIFT-SPACE][SPACE][2xSHIF
     T SPACE][COM-Q]CC[SHIFT-+]CC[SHIFT
     +]CC[SHIFT-+]CC[SHIFT-+]CC[COM-W]
1140 print"[2xSPACE]B[2xSHIFT-SPACE]B[2
     xSHIFT-SPACE]B[2xSHIFT-SPACE]B[2xS
     HIFT SPACE]B[2xSHIFT-SPACE]B[4xSPA
     CE]B[2xSHIFT-SPACE]B[2xSHIFT-SPACE
     ]B[2xSHIFT-SPACE]B[2xSHIFT-SPACE]B
     [2xSHIFT-SPACE]B
1145 print"[2xSPACE][COM-Q]CC[SHIFT-+]C
     C[SHIFT-+]CC[SHIFT-+]CC[SHIFT-+]CC
     [COM-W][4xSPACE][COM-Q]CC[SHIFT-+]
     CC[SHIFT-+]CC[SHIFT-+]CC[SHIFT-+]C
     C[COM-W]
1150 print"[2xSPACE]B[2xSHIFT-SPACE]B[2
     xSHIFT-SPACE]B[2xSHIFT-SPACE]B[2xS
     HIFT SPACE]B[2xSHIFT-SPACE]B[SHIFT
      SPACE][SPACE][2xSHIFT-SPACE]B[2xS
     HIFT SPACE]B[2xSHIFT-SPACE]B[2xSHI
     FT SPACE]B[2xSHIFT-SPACE]B[2xSHIFT
      SPACE]B
1155 print"[2xSPACE][COM-Z]CC[COM-E]CC[
     COM-E]CC[COM-E]CC[COM-E]CC[COM-X][
     4xSPACE][COM-Z]CC[COM-E]CC[COM-E]C
     C[COM-E]CC[COM-E]CC[COM-X]
1160 return
1165 rem plaatsbepaling cursor
1170 poke211,x:poke214,y:sys58640
1175 return
1180 rem lettermatrix opzetten
1185 print"[SHIFT-CLR][CRSR-DOWN][14xSP
     ACE][CTRL-9][SPACE]l[SPACE]i[SPACE
     ]n[SPACE]g[SPACE]o[SPACE][CTRL-0]"
1190 x=2:y=4:gosub1165:printsp$(1)
1195 x=38-len(sp$(2)):gosub1165:printsp
```

```
     $(2)
1200 print"[2xCRSR-DOWN][14xSPACE][COM-
     A]C[COM-R]C[COM-R]C[COM-R]C[COM-R]
     C[COM-S]
1205 print"[14xSPACE]B[SPACE]B[SHIFT-SP
     ACE]B[SHIFT-SPACE]B[SHIFT-SPACE]B[
     SHIFT-SPACE]B
1210 print"[14xSPACE][COM-Q]C[SHIFT-+]C
     [SHIFT-+]C[SHIFT-+]C[SHIFT-+]C[COM
      W]
1215 print"[14xSPACE]B[SHIFT-SPACE]B[SH
     IFT SPACE]B[SHIFT-SPACE]B[SHIFT-SP
     ACE]B[SHIFT-SPACE]B
1220 print"[14xSPACE][COM-Q]C[SHIFT-+]C
     [SHIFT-+]C[SHIFT-+]C[SHIFT-+]C[COM
      W]
1225 print"[14xSPACE]B[SHIFT-SPACE]B[SH
     IFT SPACE]B[SHIFT-SPACE]B[SHIFT-SP
     ACE]B[SHIFT-SPACE]B
1230 print"[14xSPACE][COM-Q]C[SHIFT-+]C
     [SHIFT-+]C[SHIFT-+]C[SHIFT-+]C[COM
      W]
1235 print"[14xSPACE]B[SHIFT-SPACE]B[SH
     IFT SPACE]B[SHIFT-SPACE]B[SHIFT-SP
     ACE]B[SHIFT-SPACE]B
1240 print"[14xSPACE][COM-Q]C[SHIFT-+]C
     [SHIFT-+]C[SHIFT-+]C[SHIFT-+]C[COM
      W]
1245 print"[14xSPACE]B[SHIFT-SPACE]B[SH
     IFT SPACE]B[SHIFT-SPACE]B[SHIFT-SP
     ACE]B[SHIFT-SPACE]B
1250 print"[14xSPACE][COM-Z]C[COM-E]C[C
     OM E]C[COM-E]C[COM-E]C[COM-X]
1255 return
1260 rem controle op lingo
1265 pe=p:x=0
1270 fort=0to4
1275 forn=petope+12step3
1280 ifpeek(n)<>244thenx=0:goto1295
1285 x=x+1:ifx=5thengosub1400:goto1395
1290 nextn
1295 pe=pe+80
1300 nextt
1305 pe=p:x=0
1310 fort=0to4
1315 forn=petope+400step80
1320 ifpeek(n)<>244thenx=0:goto1335
1325 x=x+1:ifx=5thengosub1455:goto1395
1330 nextn
1335 pe=pe+3
1340 nextt
1345 pe=p:x=0
1350 forn=petope+332step83
1355 ifpeek(n)<>244thenx=0:goto1365
1360 x=x+1:nextn
1365 ifx=5thengosub1505:goto1395
1370 pe=p+12:x=0
1375 forn=petope+308 step77
1380 ifpeek(n)<>244thenx=0:goto1390
1385 x=x+1:nextn
1390 ifx=5thengosub1555:goto1395
1395 return
1400 rem invullen lingo
1405 lc=1
1410 forn=petope+12step3
1415 poken,asc(mid$(l$,lc,1))-64:poken+
     1,32:lc=lc+1
1420 nextn
1425 fornn=14to2step-1
1430 forn=pe+vctope+vc+12step3
1435 poken,nn
1440 forc=0to10:next
1445 nextn,nn
1450 return
1455 lc=1
1460 forn=petope+320step80
1465 poken,asc(mid$(l$,lc,1))-64:poken+
     1,32:lc=lc+1
1470 nextn
1475 fornn=14to2step-1
1480 forn=pe+vctope+vc+320step80
1485 poken,nn
1490 forc=0to10:next
1495 nextn,nn
1500 return
1505 lc=1
1510 forn=petope+332step83
1515 poken,asc(mid$(l$,lc,1))-64:poken+
     1,32:lc=lc+1
1520 nextn
1525 fornn=14to2step-1
1530 forn=pe+vctope+vc+332step83
1535 poken,nn
1540 forc=0to10:next
1545 nextn,nn
1550 return
1555 lc=1
1560 forn=pe+308topestep-77
1565 poken,asc(mid$(l$,lc,1))-64:poken+
     1,32:lc=lc+1
1570 nextn
1575 fornn=14to2step-1
1580 forn=pe+vc+308tope+vcstep-77
1585 poken,nn
1590 forc=0to10:next
1595 nextn,nn
1600 return
1605 rem invoer
1610 print"[SHIFT-CLR][CRSR-DOWN][14xSP
     ACE][CTRL-9][SPACE]l[SPACE]i[SPACE
     ]n[SPACE]g[SPACE]o[SPACE][2xCRSR-D
     OWN]"
1615 fort=0to1000:next:poke53281,15:pok
     e53280,15
1620 print"[2xCRSR-DOWN]woorden[SPACE]i
     nvoeren[SPACE]vanaf:"
1625 print"[CRSR-DOWN][4xSPACE]1[SPACE]
     toetsenbord"
1630 print"[CRSR-DOWN][4xSPACE]2[SPACE]
     disk
1635 print"[3xCRSR-DOWN]uw[SPACE]keus[S
     PACE]is:";
1640 geta$:ifa$<"1"ora$>"2"then1640
1645 ifa$="2"thenprint"[SPACE]disk":for
     n=0to1000:next: goto1760
1650 print"[SPACE]toetsenbord"
1655 forn=0to1000:next
1660 print"[SHIFT-CLR][2xCRSR-DOWN]invo
     er[SPACE]vanaf[SPACE]toetsenbord[S
     PACE]"
1665 print"[2xCRSR-DOWN]stop[SPACE]invo
     er[SPACE]door[SPACE]*[SPACE]in[SPA
     CE]te[SPACE]tikken.[CRSR-DOWN]"
1670 print"maximaal[SPACE]100[SPACE]woo
     rden![2xCRSR-DOWN]"
1675 wc=1
```

```
1680 print"[HOME][10xCRSR-DOWN]woord[SP
     ACE]nummer:"wc;
1685 input"[8xSPACE][8xCRSR-LEFT]"; w$(
     wc)
1690 ifw$(wc)="*"thenwc=wc-1:goto1705
1695 iflen(w$(wc))<>5then1680
1700 wc=wc+1:goto1680
1705 print"[2xCRSR-DOWN]wilt[SPACE]u[SP
     ACE]spelen[SPACE]of[SPACE]de[SPACE
     ]woorden[SPACE]bewaren?"
1710 geta$:ifa$<>"s"anda$<>"b"then1710
1715 ifa$="s"then return
1720 input"[2xCRSR-DOWN]geef[SPACE]file
     [SPACE]naam";f$
1725 open1,8,14,"`0:"+f$+",s,w"
1730 print#1,wc;chr$(13)
1735 fort=1towc
1740 print#1,w$(t)
1745 next t:close1
1750 printchr$(147)
1755 return
1760 print"[SHIFT-CLR][2xCRSR-DOWN]invo
     er[SPACE]vanaf[SPACE]disk"
1765 input"[3xCRSR-DOWN]geef[SPACE]file
     [SPACE]naam";f$
1770 open15,8,15
1775 open1,8,14,"`0:"+f$+",s,r"
1780 input#15,a$,b$,c$,d$
1785 ifval(a$)<>62then1810
1790 print"[4xCRSR-DOWN]";f$;"[SPACE]ni
     et[SPACE]gevonden!!!![6xCRSR-DOWN]
     "
1795 print"druk[SPACE]op[SPACE]de[SPACE
     ]spatiebalk."
1800 geta$:ifa$<>"[SPACE]"then1800
1805 close1:close15:    goto1605
1810 input#1,wc
1815 fort=1towc
1820 input#1,w$(t)
1825 nextt:close1
1830 close15
1835 return
1840 rem titelscherm
1845 print"[SHIFT-CLR]":poke53281,3:pok
     e53280,3:poke646,0
1850 print"[2xSPACE][2xCOM-+]"
1855 print"[2xSPACE][2xCOM-+][12xSPACE]
     e[SPACE]e[SPACE]n"
1860 print"[2xSPACE][2xCOM-+]"
1865 print"[2xSPACE][2xCOM-+][6xSPACE]w
     [SPACE]o[SPACE]o[SPACE]r[SPACE]d[S
     PACE]s[SPACE]p[SPACE]e[SPACE]l[8xS
     PACE]"
1870 print"[2xSPACE][2xCOM-+]"
1875 print"[2xSPACE][2xCOM-+]"
1880 print"[2xSPACE][2xCOM-+]"
1885 print"[2xSPACE][2xCOM-+][3xSPACE][
     2xCOM-+][2xSPACE][2xCOM-+][4xSPACE
     ][2xCOM-+][2xSPACE][6xCOM-+][2xSPA
     CE][6xCOM-+]"
1890 print"[2xSPACE][2xCOM-+][3xSPACE][
     2xCOM-+][2xSPACE][3xCOM-+][3xSPACE
     ][2xCOM-+][2xSPACE][2xCOM-+][3xSPA
     CE][COM-+][2xSPACE][2xCOM-+][2xSPA
     CE][2xCOM-+]"
1895 print"[2xSPACE][2xCOM-+][3xSPACE][
     2xCOM-+][2xSPACE][4xCOM-+][2xSPACE
     ][2xCOM-+][2xSPACE][2xCOM-+][3xSPA
     CE][COM-+][2xSPACE][2xCOM-+][2xSPA
     CE][2xCOM-+]"
1900 print"[2xSPACE][2xCOM-+][3xSPACE][
     2xCOM-+][2xSPACE][2xCOM-+][SPACE][
     2xCOM-+][SPACE][2xCOM-+][2xSPACE][
     2xCOM-+][6xSPACE][2xCOM-+][2xSPACE
     ][2xCOM-+]"
1905 print"[2xSPACE][2xCOM-+][3xSPACE][
     2xCOM-+][2xSPACE][2xCOM-+][2xSPACE
     ][4xCOM-+][2xSPACE][2xCOM-+][2xSPA
     CE][2xCOM-+][2xSPACE][2xCOM-+][2xS
     PACE][2xCOM-+]"
1910 print"[2xSPACE][2xCOM-+][3xSPACE][
     2xCOM-+][2xSPACE][2xCOM-+][3xSPACE
     ][3xCOM-+][2xSPACE][2xCOM-+][3xSPA
     CE][COM-+][2xSPACE][2xCOM-+][2xSPA
     CE][2xCOM-+]"
1915 print"[2xSPACE][2xCOM-+][3xSPACE][
     2xCOM-+][2xSPACE][2xCOM-+][4xSPACE
     ][2xCOM-+][2xSPACE][6xCOM-+][2xSPA
     CE][6xCOM-+]"
1920 print"[2xSPACE][2xCOM-+]"
1925 print"[2xSPACE][2xCOM-+]"
1930 print"[2xSPACE][33xCOM-+]"
1935 print"[2xSPACE][33xCOM-+][2xCRSR-D
     OWN]"
1940 print"[16xSPACE]door"
1945 print"
1950 print"[12xSPACE]s.f.[SPACE]de[SPAC
     E]vries"
1955 return
1960 rem plaatsen mach. taal
1965 fort=49152to49292:readq:poket,q
1970 tq=tq+q
1975 nextt
1980 iftq<>15295thenprint"fout[SPACE]in
     [SPACE]data":end
1985 return
1990 rem data mach. taal
1995 data162,0,189,0,4,157,0,194
2000 data189,0,5,157,0,195,189,0
2005 data6,157,0,196,189,0,7,157
2010 data0,197,232,208,229,96,162,0
2015 data189,0,4,157,0,198,189,0
2020 data5,157,0,199,189,0,6,157
2025 data0,200,189,0,7,157,0,201
2030 data232,208,229,96,162,0,189,0
2035 data194,157,0,4,189,0,195,157
2040 data0,5,189,0,196,157,0,6
2045 data189,0,197,157,0,7,232,208
2050 data229,76,121,192,162,0,189,0
2055 data198,157,0,4,189,0,199,157
2060 data0,5,189,0,200,157,0,6
2065 data189,0,201,157,0,7,232,208
2070 data229,169,0,162,0,157,0,216
2075 data157,0,217,157,0,218,157,0
2080 data219,232,208,241,96
2085 rem uitleg spel
2090 print"[SHIFT-CLR][CTRL-8]";
2095 print"alles[SPACE]wat[SPACE]je[SPA
     CE]moet[SPACE]doen[SPACE]wordt[SPA
     CE]op[SPACE]het[5xSPACE]scherm[SPA
     CE]aangegeven,daarom[SPACE]";
2100 print"alleen[SPACE]de[6xSPACE]volg
     ende[SPACE]opmerkingen:[CRSR-DOWN]
     "
2105 print"1.[SPACE]de[SPACE]bedenktijd
     [SPACE]is[SPACE]beperkt."
```

```
2110 print"2.[SPACE]druk[SPACE]niet[SPA
     CE]op[SPACE][CTRL-9]return[CTRL-0]
     [SPACE]na[SPACE]invullen[SPACE]van
     [5xSPACE]het[SPACE]antwoord."
2115 print"3.[SPACE]correctie[SPACE]van
     [SPACE]het[SPACE]antwoord[SPACE]is
     [SPACE]niet[6xSPACE]mogelijk."
2120 print"4.[SPACE]wacht[SPACE]je[SPAC
     E]te[SPACE]lang,[SPACE]dan[SPACE]g
     aat[SPACE]de[SPACE]beurt[5xSPACE]o
     ver[SPACE]naar[SPACE]je[SPACE]tege
     nspeler.
2125 print"5.[SPACE]raad[SPACE]je[SPACE
     ]het[SPACE]woord[SPACE]niet[SPACE]
     in[SPACE]5[SPACE]keer[SPACE]dan[4x
     SPACE]gaat[SPACE]de[SPACE]beurt[SP
     ACE]ook[SPACE]over.
2130 print"6.[SPACE]wachten[SPACE]beide
     [SPACE]spelers[SPACE]te[SPACE]lang
     [SPACE]of[8xSPACE]wordt[SPACE]het[
     SPACE]woord[SPACE]niet[SPACE]gerad
     en"
2135 print"[3xSPACE]dan[SPACE]vult[SPAC
     E]de[SPACE]comp.[SPACE]het[SPACE]w
     oord[SPACE]in."
2140 print"7.[SPACE]elke[SPACE]speler[S
     PACE]krijgt[SPACE]naast[SPACE]de[S
     PACE]cijfer[6xSPACE]blokken[SPACE]
     3[SPACE]rode[SPACE]blokken."
2145 print"[3xSPACE]pak[SPACE]je[SPACE]
     een[SPACE]rood[SPACE]blok,[SPACE]d
     an[SPACE]gaat[SPACE]de[7xSPACE]beu
     rt[SPACE]over[SPACE]naar[SPACE]teg
     enspeler."
2150 print"8.[SPACE]5[SPACE]blokken[SPA
     CE]op[SPACE]een[SPACE]rij[SPACE]ge
     eft[SPACE][CTRL-9]lingo[CTRL-0][SP
     ACE]en[5xSPACE]levert[SPACE]50[SPA
     CE]punten[SPACE]op."
2155 print"[CRSR-DOWN][2xSPACE]druk[SPA
     CE]op[SPACE]spatiebalk"
2160 geta$:ifa$<>"[SPACE]"then2160
2165 poke646,0:return
```

** EINDE LISTING lingo **

Checksum lingo

REGEL 5	199	REGEL 250	221	REGEL 495	156	REGEL 740	209	REGEL 985	236
REGEL 10	202	REGEL 255	169	REGEL 500	63	REGEL 745	177	REGEL 990	111
REGEL 15	230	REGEL 260	242	REGEL 505	168	REGEL 750	113	REGEL 995	190
REGEL 20	175	REGEL 265	139	REGEL 510	28	REGEL 755	190	REGEL 1000	139
REGEL 25	21	REGEL 270	151	REGEL 515	153	REGEL 760	123	REGEL 1005	44
REGEL 30	83	REGEL 275	19	REGEL 520	181	REGEL 765	57	REGEL 1010	176
REGEL 35	80	REGEL 280	180	REGEL 525	132	REGEL 770	25	REGEL 1015	38
REGEL 40	54	REGEL 285	97	REGEL 530	118	REGEL 775	221	REGEL 1020	236
REGEL 45	227	REGEL 290	57	REGEL 535	199	REGEL 780	169	REGEL 1025	143
REGEL 50	90	REGEL 295	39	REGEL 540	61	REGEL 785	75	REGEL 1030	174
REGEL 55	188	REGEL 300	143	REGEL 545	29	REGEL 790	37	REGEL 1035	105
REGEL 60	223	REGEL 305	146	REGEL 550	160	REGEL 795	177	REGEL 1040	80
REGEL 65	93	REGEL 310	151	REGEL 555	126	REGEL 800	244	REGEL 1045	43
REGEL 70	119	REGEL 315	63	REGEL 560	103	REGEL 805	32	REGEL 1050	204
REGEL 75	115	REGEL 320	8	REGEL 565	112	REGEL 810	190	REGEL 1055	65
REGEL 80	113	REGEL 325	121	REGEL 570	205	REGEL 815	25	REGEL 1060	208
REGEL 85	13	REGEL 330	96	REGEL 575	83	REGEL 820	221	REGEL 1065	116
REGEL 90	92	REGEL 335	248	REGEL 580	196	REGEL 825	214	REGEL 1070	99
REGEL 95	89	REGEL 340	145	REGEL 585	33	REGEL 830	143	REGEL 1075	207
REGEL 100	55	REGEL 345	210	REGEL 590	99	REGEL 835	174	REGEL 1080	167
REGEL 105	200	REGEL 350	77	REGEL 595	228	REGEL 840	224	REGEL 1085	41
REGEL 110	74	REGEL 355	125	REGEL 600	231	REGEL 845	208	REGEL 1090	46
REGEL 115	125	REGEL 360	194	REGEL 605	73	REGEL 850	116	REGEL 1095	101
REGEL 120	185	REGEL 365	149	REGEL 610	13	REGEL 855	120	REGEL 1100	47
REGEL 125	50	REGEL 370	110	REGEL 615	36	REGEL 860	38	REGEL 1105	101
REGEL 130	28	REGEL 375	134	REGEL 620	151	REGEL 865	57	REGEL 1110	83
REGEL 135	193	REGEL 380	139	REGEL 625	79	REGEL 870	137	REGEL 1115	139
REGEL 140	62	REGEL 385	210	REGEL 630	151	REGEL 875	224	REGEL 1120	83
REGEL 145	28	REGEL 390	137	REGEL 635	29	REGEL 880	101	REGEL 1125	139
REGEL 150	213	REGEL 395	133	REGEL 640	117	REGEL 885	190	REGEL 1130	19
REGEL 155	163	REGEL 400	101	REGEL 645	199	REGEL 890	88	REGEL 1135	107
REGEL 160	215	REGEL 405	145	REGEL 650	163	REGEL 895	34	REGEL 1140	83
REGEL 165	27	REGEL 410	208	REGEL 655	5	REGEL 900	224	REGEL 1145	139
REGEL 170	55	REGEL 415	114	REGEL 660	8	REGEL 905	176	REGEL 1150	51
REGEL 175	224	REGEL 420	110	REGEL 665	151	REGEL 910	9	REGEL 1155	83
REGEL 180	249	REGEL 425	145	REGEL 670	225	REGEL 915	236	REGEL 1160	142
REGEL 185	180	REGEL 430	62	REGEL 675	33	REGEL 920	143	REGEL 1165	116
REGEL 190	18	REGEL 435	128	REGEL 680	123	REGEL 925	174	REGEL 1170	123
REGEL 195	183	REGEL 440	117	REGEL 685	126	REGEL 930	60	REGEL 1175	142
REGEL 200	230	REGEL 445	249	REGEL 690	174	REGEL 935	44	REGEL 1180	173
REGEL 205	143	REGEL 450	113	REGEL 695	246	REGEL 940	43	REGEL 1185	158
REGEL 210	125	REGEL 455	123	REGEL 700	36	REGEL 945	204	REGEL 1190	101
REGEL 215	148	REGEL 460	197	REGEL 705	121	REGEL 950	65	REGEL 1195	47
REGEL 220	157	REGEL 465	2	REGEL 710	105	REGEL 955	208	REGEL 1200	210
REGEL 225	78	REGEL 470	114	REGEL 715	129	REGEL 960	116	REGEL 1205	199
REGEL 230	127	REGEL 475	105	REGEL 720	254	REGEL 965	97	REGEL 1210	84
REGEL 235	129	REGEL 480	28	REGEL 725	106	REGEL 970	185	REGEL 1215	103
REGEL 240	213	REGEL 485	68	REGEL 730	143	REGEL 975	157	REGEL 1220	84
REGEL 245	149	REGEL 490	89	REGEL 735	128	REGEL 980	38	REGEL 1225	103

Checksum Lingo (vervolg)

REGEL 1230	84	REGEL 1420	208	REGEL 1610	46	REGEL 1800	72	REGEL 1990	18
REGEL 1235	103	REGEL 1425	143	REGEL 1615	163	REGEL 1805	160	REGEL 1995	241
REGEL 1240	84	REGEL 1430	190	REGEL 1620	41	REGEL 1810	123	REGEL 2000	252
REGEL 1245	103	REGEL 1435	173	REGEL 1625	104	REGEL 1815	246	REGEL 2005	0
REGEL 1250	184	REGEL 1440	103	REGEL 1630	41	REGEL 1820	1	REGEL 2010	142
REGEL 1255	142	REGEL 1445	152	REGEL 1635	5	REGEL 1825	225	REGEL 2015	254
REGEL 1260	13	REGEL 1450	142	REGEL 1640	6	REGEL 1830	6	REGEL 2020	1
REGEL 1265	11	REGEL 1455	114	REGEL 1645	100	REGEL 1835	142	REGEL 2025	226
REGEL 1270	143	REGEL 1460	159	REGEL 1650	38	REGEL 1840	211	REGEL 2030	143
REGEL 1275	56	REGEL 1465	80	REGEL 1655	210	REGEL 1845	162	REGEL 2035	100
REGEL 1280	95	REGEL 1470	208	REGEL 1660	26	REGEL 1850	41	REGEL 2040	145
REGEL 1285	207	REGEL 1475	143	REGEL 1665	171	REGEL 1855	1	REGEL 2045	95
REGEL 1290	208	REGEL 1480	37	REGEL 1670	25	REGEL 1860	41	REGEL 2050	140
REGEL 1295	238	REGEL 1485	173	REGEL 1675	125	REGEL 1865	232	REGEL 2055	108
REGEL 1300	214	REGEL 1490	103	REGEL 1680	8	REGEL 1870	41	REGEL 2060	131
REGEL 1305	11	REGEL 1495	152	REGEL 1685	82	REGEL 1875	41	REGEL 2065	81
REGEL 1310	143	REGEL 1500	142	REGEL 1690	10	REGEL 1880	41	REGEL 2070	83
REGEL 1315	158	REGEL 1505	114	REGEL 1695	20	REGEL 1885	213	REGEL 2075	83
REGEL 1320	90	REGEL 1510	165	REGEL 1700	83	REGEL 1890	61	REGEL 2080	6
REGEL 1325	217	REGEL 1515	80	REGEL 1705	218	REGEL 1895	227	REGEL 2085	141
REGEL 1330	208	REGEL 1520	208	REGEL 1710	153	REGEL 1900	61	REGEL 2090	73
REGEL 1335	185	REGEL 1525	143	REGEL 1715	110	REGEL 1905	137	REGEL 2095	226
REGEL 1340	214	REGEL 1530	43	REGEL 1720	228	REGEL 1910	61	REGEL 2100	245
REGEL 1345	11	REGEL 1535	173	REGEL 1725	227	REGEL 1915	213	REGEL 2105	112
REGEL 1350	165	REGEL 1540	103	REGEL 1730	70	REGEL 1920	41	REGEL 2110	36
REGEL 1355	93	REGEL 1545	152	REGEL 1735	246	REGEL 1925	41	REGEL 2115	94
REGEL 1360	71	REGEL 1550	142	REGEL 1740	21	REGEL 1930	67	REGEL 2120	94
REGEL 1365	94	REGEL 1555	114	REGEL 1745	225	REGEL 1935	101	REGEL 2125	162
REGEL 1370	24	REGEL 1560	86	REGEL 1750	77	REGEL 1940	17	REGEL 2130	52
REGEL 1375	171	REGEL 1565	80	REGEL 1755	142	REGEL 1945	187	REGEL 2135	18
REGEL 1380	91	REGEL 1570	208	REGEL 1760	252	REGEL 1950	228	REGEL 2140	67
REGEL 1385	71	REGEL 1575	143	REGEL 1765	245	REGEL 1955	142	REGEL 2145	109
REGEL 1390	99	REGEL 1580	220	REGEL 1770	251	REGEL 1960	80	REGEL 2150	211
REGEL 1395	142	REGEL 1585	173	REGEL 1775	222	REGEL 1965	238	REGEL 2155	163
REGEL 1400	117	REGEL 1590	103	REGEL 1780	52	REGEL 1970	247	REGEL 2160	72
REGEL 1405	114	REGEL 1595	152	REGEL 1785	67	REGEL 1975	214	REGEL 2165	91
REGEL 1410	56	REGEL 1600	142	REGEL 1790	181	REGEL 1980	199		
REGEL 1415	80	REGEL 1605	98	REGEL 1795	73	REGEL 1985	142		

REKENWONDER

Enkele algemene opmerkingen en tips

- Wanneer men cijfergegevens dient in te voeren moet men er rekening mee houden dat de computer bij het geven van een uitkomst, achter deze uitkomst automatisch de grondeenheid uit het SI-stelsel (System International) zal plaatsen, d.w.z.: m (meter), mÇU2\ (vierkante meter),... Wanneer men nu een lengte van 23 cm heeft, kan men ofwel '0.23' intypen (in meter dus), ofwel geeft men gewoon '23' in, maar houdt men er rekening mee dat de eenheid achter de uitkomst niet 'meter' moet zijn, maar 'centimeter' ! Tip: 1 m = 1000 dm = 1000000 cm.
- Het getal 'Pi' is in dit programma gelijk aan 3.14159265 !!
- Gebruik bij het invoeren van cijfergegevens NOOIT een komma, maar wel een punt ! Dus: '24.89' i.p.v. '24,89' !!
- Er wordt in dit programma geen gebruik gemaakt van de diameter, maar wel van de straal (diameter = 2 x straal).
- Als men ergens een keuze moet maken, staan de keuzemogelijkheden tussen haakjes achter de vraag. E}n van deze mogelijkheden moet dan, zonder de 'SHIFT'-toets, ingetypt worden. In de menu's (hoofdmenu's en andere) moet gewoon op de gewenste cijfer- of functietoets gedrukt worden. Alleen bij cijfergegevens is een druk op de 'RETURN'-toets nodig !
- Wanneer een uitkomst als begin '0,...' heeft, zal de computer deze '0' niet tonen, enkel een '.' als komma en daarna de getallen achter de komma ! Bij het invoeren van gegevens echter zal de computer bv. '0.623' wel aanvaarden !!
- In verschillende gevallen mag een waarde niet gelijk zijn aan '0', omdat de computer dan moet delen door nul (wat wiskundig niet mag). De computer zal dan de vraag (of vragen) herhalen. Ook wanneer om bepaalde andere redenen een invoer niet mogelijk is zal de computer dit melden en de vraag (of vragen) herhalen !
- Het indrukken van de 'RETURN'-toets als men getallen moet invoeren, heeft hetzelfde resultaat als het indrukken van de '0'-toets !!

GEBRUIKTE AFKORTINGEN :

S = Oppervlakte (Surface).
r = Straal.
m = Meter.
m2 = Vierkante meter.
m3 = Kubieke meter.
b = Basis (Gebruikt bij de piramide = lengte van 1 zijde v.h.
grondvlak).

z = Zijde (Gebruikt bij de piramide = afstand v.d. top v.d.
piramide tot een hoekpunt v.h. grondvlak).
V = Volume.
h = Hoogte.
C = Celsius.
F = Fahrenheit.
% = Procent.

Instrukties

Na een korte inleiding tot het programma, verschijnt het MENU op het scherm. Men kan tussen 8 verschillende mogelijkheden kiezen. Telkens nadat de computer gedaan heeft met iets te berekenen uit ingevoerde gegevens dient men op de 'RETURN'-toets te drukken om terug naar het hoofdmenu te keren. In dit menu heeft men ook steeds de mogelijkheid om het programma te verlaten en terug te keren naar BASIC. Hiervoor dient de 'SHIFT'-toets samen met de 'RETURN'-toets ingedrukt te worden.
Nadat men }}n van de nummers van 1 t/m 8 heeft ingedrukt, verschijnt de titel achter dit nummer bovenaan het scherm in een andere kleur, waarna men het programma echt kan gaan gebruiken !!

Bespreking

1. Vierkantsvergelijkingen.
Na '1' ingedrukt te hebben heeft men de mogelijkheid om vierkantsvergelijkingen (of tweedegraadsvergelijkingen) op te lossen van het type axÇU2\+bx+c=0. Dit is de vergelijking van een parabool in de wiskunde. Door de waarden van a, b en c in te voeren, kan de computer de overblijvende 'x' berekenen. Deze 'x' is het snijpunt van de parabool met de X-as van het assenkruis. Meestal heeft de parabool geen of 2 snijpunten met de X-as, maar het is mogelijk dat hij slechts in }}n punt raakt. De x-en worden de wortels van de vergelijking genoemd ! Wanneer de waarde van 'a' nul is, heeft men geen vergelijking van een parabool meer, maar wel de vergelijking van een rechte (eerste@graadsvergelijking van de vorm ax+b=0). Wanneer 'a' EN 'b' gelijk zijn aan nul, moet 'c' ook nul zijn. Men heeft dan een identieke vergelijking. Is 'c' echter niet gelijk aan nul, dan heeft men een valse vergelijking, want als 'c' bijvoorbeeld '3' is krijgt men in de vergelijking '3=0', wat natuurlijk niet kan !!
Verder ga ik op dit deel van het programma niet in, omdat men hierover leert in het hoger middelbaar onderwijs en de volwassenen hiermee meestal niets kunnen doen in het dagelijkse leven. Voor de studenten echter kan het wel gemakkelijk en nuttig zijn, omdat het zelf berekenen van de wortels van zo'n vergelijking toch wat tijd in beslag neemt.

2. Oppervlakte- en volumeberekening.
Als men in het hoofdmenu '2' gekozen heeft, komt men in een ander menu terecht, waar men kan kiezen tussen oppervlakte- en volumeberekening van bepaalde figuren. Omdat de oppervlakte en het volume van een vierkant (kubus), rechthoek (balk),... gemakkelijk te berekenen zijn, zijn in dit programma enkel de moeilijkere figuren voorzien, nl. piramide, kegel, cirkel en bol.
Nadat men }}n van 5 mogelijkheden gekozen heeft, wordt er meestal nog eens de vraag gesteld: "Wat is er gevraagd ?", met daarna enkele keuzemogelijkheden. Het eerste hiervan is steeds oppervlakte (S) of volume (V), maar er is ook de mogelijkheid ingebouwd om bijvoorbeeld de straal te berekenen van een bol, wanneer de oppervlakte of het volume reeds gegeven zijn.
Wanneer men nu }}n van de mogelijkheden tussen haakjes (in KLEINE letter, dus zonder 'SHIFT') indrukt, worden er 1 of meerdere gegevens gevraagd, waarna de computer het gewenste zal berekenen.
Enkele opmerkingen :

- Bij '3' (= oppervlakte piramide) wordt onmiddellijk naar de waarde van 'b' en 'z' gevraagd, dit zijn de basis van de piramide, en de afstand tussen de top van de piramide en }}n van de hoekpunten van zijn grondvlak (zie afkortingen). Let er wel op dat het hier gaat om een piramide met een VIERKANT als grondvlak !!
- Bij '5' (= volume piramide-kegel) wordt er rechtstreeks gebruik gemaakt van de oppervlakte van het grondvlak (van om het even welke regelmatige veelhoek of een cirkel) om het volume en de hoogte te berekenen ! De oppervlakte van het grondvlak van de kegel kan men gaan berekenen bij '1' (= oppervlakte cirkel) !!

3. Procent- en gemiddeldeberekening.
Wanneer men in het menu '3' gekozen heeft, kan men (nu met de functietoetsen) kiezen tussen procentberekening of gemiddeldebe@rekening.
Bij procentberekening moet ingegeven worden hoeveel het maximum is (van bv. te behalen punten van een leerling). Daarna wordt er gevraagd hoeveel er behaald is. Zo kan de computer berekenen hoeveel procent het 2de van het 1ste getal uitmaakt !
Bij gemiddeldeberekening moet men eerst ingeven van hoeveel verschillende getallen men het gemiddelde wil berekenen. Wanneer men hier '0' intypt (gevolgd door 'RETURN') zal de computer terugkeren naar het hoofdmenu. Indien een ander getal wordt ingegeven, vraagt de computer dat aantal getallen in te voeren (met natuurlijk steeds 'RETURN' ertussen). Als het laatste getal is ingevoerd, geeft de computer onmiddellijk het gemiddelde van al de getallen.

4. Celsius Fahrenheit.
Weer moet men met behulp van de functietoetsen een keuze maken. Door middel van 'f1' vraagt de computer naar een aantal graden Celsius en berekent hoeveel dit in graden Fahrenheit is, en met 'f3' vraagt de computer een aantal gra-

den Fahrenheit en berekent hoeveel dit in graden Celsius is. Opmerking: -40{Celsius = -40{Fahrenheit !!

5. Opbrengst van een kapitaal.

Als in het hoofdmenu '5' gekozen werd krijgt men nog een klein menu waarin gevraagd wordt wat men zou willen berekenen. Wanneer men '1' kiest, kan men bijvoorbeeld berekenen hoeveel een bedrag van 10000 fr. na 10 jaar opbrengt als men er bij de bank 5.5 % rente voor krijgt (17081.4 fr.). Ook interessant is dat men kan berekenen hoeveel jaar het duurt vooraleer dit bedrag bijvoorbeeld verdubbeld is, met 5.5 % rente (bijna 13 jaar !). Om dit te berekenen, moet men in het menu van "Opbrengst van een kapitaal" nummer '4' kiezen. Daarna vraagt de computer begin- en eindkapitaal (bv. 10000 en 20000 fr.), en de waarde van de rente (bv 5.5 %). Als dit alles is ingevoerd geeft de computer het resultaat. Zo kan men ook nog een beginkapitaal en een rente berekenen, als de 3 andere faktoren telkens gekend zijn ! (Opmerking: voor 'guldens' blijven alle berekeningen natuurlijk juist dezelfde, en is het programma ook te gebruiken, alleen wordt dan achter een resultaat foutief 'frank' geplaatst).

6. Wortels en machten.

Door in het hoofdmenu '6' te kiezen kan men weer 2 dingen gaan doen: worteltrekkingen of machten. Bij worteltrekkingen moet de machtswortel ingegeven worden, en uit welk getal deze moet getrokken worden. Zo is bijvoorbeeld de 3de machtswortel uit 125 gelijk aan 5 (5x5x5=125). Als men in plaats van f1, f3 kiest, kan men machten gaan berekenen. De computer vraagt de macht en het getal en geeft daarop de uitkomst. Zo is bijvoorbeeld de 2.5de macht van 2.4 gelijk aan 8.92335364 !

7. Sinus, Cosinus, Tangens,...

Wanneer men in het hoofdmenu '7' ingedrukt heeft, heeft men de mogelijkheid om ofwel van een hoek de goniometrische getallen (sinus, cosinus, tangens, cosecans, secans en cotangens) te vragen. Men kan (met behulp van f3) ook een sinus, cosinus of tangens ingeven, waarna de computer de hoek geeft. Normaal werkt de Commodore 64 in radialen, maar in dit programma is het mogelijk gemaakt om de hoek gewoon in graden op te geven en toch de juiste goniometrische getallen te bekomen. Ook wanneer men sinus, cosinus of tangens ingevoerd heeft, wordt de uitkomst (de hoek dus) ook in graden weergegeven en niet in radialen !!

8. Weekdagen bepalen en priemgetallen.

BESPREKING :

1) Weekdagen bepalen.

Hiermee kan men te weten komen op welke dag bijvoorbeeld de 1ste januari van het jaar 1992 valt (Woensdag). Men heeft 2 mogelijkheden om de datum in te geven. Ofwel geeft men dag, maand en jaar in gescheiden door komma's (bv. 01,01,92 (of ook: 1,1,92)), ofwel drukt men telkens op 'RETURN', in plaats van op de komma, dus: 1 (RETURN) 1 (RETURN) 92 (RETURN). Wat niet mag is het ingeven van een jaar bestaande uit 4 cijfers, dus niet '1992', maar wel '92' (deze "kalender" is dus geldig tot het jaar 2000). Wanneer een niet bestaande datum wordt ingevoerd, wordt men hierop attent gemaakt door de computer !

3) Priemgetal of niet ?!

Hiermee kan men te weten komen of een getal al of niet een priemgetal is. Een priemgetal is een getal dat uitsluitend deelbaar is door }}n en zichzelf ! Is het ingevoerde getal geen priemgetal dan geeft de computer de kleinste deler ('1' niet meegeteld).

5) Alle priemgetallen tot...

Wanneer 'f5' gekozen wordt vraagt de computer tot welk getal hij de priemgetallen moet berekenen. Wanneer men bijvoorbeeld '103' indrukt (gevolgd door 'RETURN') geeft de computer alle priemgetallen tot en met '103'. In basic kunnen deze berekeningen wel even duren, maar in machinetaal zijn ze snel berekent !!

Uitleg bij het programma 'REKENWONDER'.

10	Run/stop, Restore, List, Save en Shift + com.-toets uitschakelen + toetsen repeterend maken.
20-90	Kleine introductie.
100-130	Titelscherm (+ 'CLR').
140-260	Hoofdmenu.
270-510	Vierkantsvergelijkingen oplossen.
520-1000	Oppervlakte- en volumeberekening :
520-620	Menu.
630-690	Oppervlakte cirkel.
700-760	Oppervlakte bol.
770-820	Oppervlakte piramide.
830-930	Volume piramide-kegel.
940-1000	Volume bol.
1010-1190	Procent- en gemiddeldeberekening :
1010-1080	Menu.
1090-1120	Procentberekening.
1130-1190	Gemiddeldeberekening.
1200-1310	Celsius Fahrenheit :
1200-1270	Menu.
1280-1290	Celsius - Fahrenheit.
1300-1310	Fahrenheit - Celsius.
1320-1470	Opbrengst van een kapitaal :
1320-1390	Menu.
1400-1410	Eindkapitaal berekenen.
1420-1430	Beginkapitaal berekenen.
1440-1450	Intrest berekenen.
1460-1470	Aantal jaar berekenen.
1480-1610	Wortels en machten :
1480-1540	Menu.
1550-1580	Worteltrekkingen.
1590-1610	Machten.
1620-2010	Sinus, Cosinus, Tangens,... :
1620-1690	Menu.
1700-1810	Hoek - Goniom. getallen.
1820-2010	Goniom. getallen - Hoek.

2020-2680	Weekdagen bepalen en priemgetallen :
2020-2100	Menu.
2110-2250	Alle priemget. tot...
2260-2390	Priemgetal of niet ?!
2400-2680	Weekdagen bepalen.
2690-2710	Resetten van het programma.
2720-2860	Goto- en gosub-routines.
2870-2900	Terugkeren naar het hoofdmenu.
2910	Wachtlus.
2920	END

```
1    rem *****************
2    rem ** rekenwonder **
3    rem *****************
4    rem geschreven door tim verlinden
5    rem ******************************
6    rem
7    rem heist op den berg
8    rem belgie
9    rem ******************************
10   poke808,254:poke657,128:poke774,0:
     poke818,32:poke792,193:poke650,128
20   dima(100),c(100):poke53280,4:poke5
     3281,14:a$="[SPACE]tim[SPACE]prese
     nteert[SPACE]":print"[CTRL-7][SHIF
     T CLR]"
30   foru=1to17:a(u)=abs(asc(mid$(a$,u,
     1))-64):next:a=1
40   c(a)=int(17*rnd(1))+1:ifa=1then70
50   foru=1toa-1:ifc(a)<>c(u)thennext:g
     oto70
60   goto40
70   poke1514+c(a),a(c(a)):a=a+1:ifa=le
     n(a$)+1then90
80   goto40
90   gosub2910:foru=55787to55803:pokeu,
     1:next:gosub2910
100  poke53280,5:poke53281,7:print"[SHI
     FT CLR][12xCRSR-DOWN]"
110  printtab(14)"[CTRL-3]rekenwonder":
     printtab(14)"[COM-4]DDDDDDDDDDD[9x
     CRSR-DOWN]"
120  print"[2xSPACE][COM-1][CTRL-9]copy
     right[SPACE]1989[SPACE]-[SPACE]tim
     ."
130  gosub2910:gosub2910:poke53280,2:po
     ke53281,6
140  print"[SHIFT-CLR]"tab(18)"[CTRL-2]
     [CTRL-9][CRSR-DOWN]MENU[2xCRSR-DOW
     N]":clr:print"[CTRL-N][SPACE]1[SPA
     CE]=[SPACE]Vierkantsvergelijkingen
     .[CRSR-DOWN]"
150  print"[SPACE]2[SPACE]=[SPACE]Opper
     vlakte-[SPACE]en[SPACE]volumeberek
     ening.[CRSR-DOWN]"
160  print"[SPACE]3[SPACE]=[SPACE]Proce
     nt-[SPACE]en[SPACE]gemiddeldeberek
     ening.[CRSR-DOWN]"
170  print"[SPACE]4[SPACE]=[SPACE]Celsi
     us[SPACE]<->[SPACE]Fahrenheit.[CRS
     R-DOWN]"
180  print"[SPACE]5[SPACE]=[SPACE]Opbre
     nst[SPACE]van[SPACE]een[SPACE]kapi
     taal.[CRSR-DOWN]"
190  print"[SPACE]6[SPACE]=[SPACE]Worte
     ls[SPACE]en[SPACE]machten.[CRSR-DO
     WN]"
200  print"[SPACE]7[SPACE]=[SPACE]Sinus
     ,[SPACE]Cosinus,[SPACE]Tangens,...
     [CRSR-DOWN]"
210  print"[SPACE]8[SPACE]=[SPACE]Weekd
     agen[SPACE]bepalen[SPACE]en[SPACE]
     priemgetallen."
220  print"[CRSR-DOWN][CRSR-RIGHT][CTRL
      9]'Shift[SPACE]+[SPACE]Return'[SP
     ACE]=[SPACE]EXIT[CRSR-UP]":printta
     b(10)"[2xCRSR-DOWN]***************
     *****"
230  getj$:ifj$=""then230
240  j=val(j$):on j goto 270,520,1010,1
     200,1320,1480,1620,2020
250  ifj$=chr$(141)then2690
260  goto140
270  print"[SHIFT-CLR][CRSR-DOWN][COM-8
     ][8xCRSR-RIGHT]Vierkantsvergelijki
     ngen."
280  printtab(8)"----------------------
     --"
290  print"[CTRL-2][CRSR-DOWN]Oplossen[
     SPACE]van[SPACE]vierkantsvergelijk
     ingen"
300  print"van[SPACE]het[SPACE]type:[SP
     ACE][2xCRSR-DOWN]ax[CRSR-UP]2[CRSR
     -DOWN]+bx+c=0":print
310  print"Voer[SPACE]de[SPACE]coeffici
     enten[SPACE]in:":print
320  input"[SPACE]a[SPACE]=";a:input"[S
     PACE]b[SPACE]=";b:input"[SPACE]c[S
     PACE]=";c
330  print"[SHIFT-CLR][CRSR-DOWN]De[SPA
     CE]vergelijking[SPACE]";a;"[CRSR-L
     EFT]x[CRSR-UP]2[CRSR-DOWN][SPACE]+
     ";b;"[CRSR-LEFT]x[SPACE]+";c;"=[SP
     ACE]0[CRSR-DOWN]":ifa=0then380
340  d=b*b-4*a*c:ifd=0then370
350  ifd<0then420
360  x1=(-b-sqr(d))/(2*a):x2=(-b+sqr(d)
     )/(2*a):goto440
370  x=-b/(2*a):goto460
380  ifb=0then400
390  x=-c/b:goto480
400  ifc=0then510
410  goto500
420  print"[COM-7]Heeft[SPACE]geen[SPAC
     E]reele[SPACE]wortels[SPACE]!!"
430  print"[CTRL-2][CRSR-DOWN]De[SPACE]
     discriminant[SPACE]is:[COM-7][SPAC
     E]";d"[CTRL-2]":goto2870
440  print"Heeft[SPACE]als[SPACE]wortel
     s:[SPACE][CRSR-DOWN]"
450  print"[COM-7]x1[SPACE]=[SPACE]";x1
     :print"[CRSR-DOWN]x2[SPACE]=[SPACE
     ]";x2"[CTRL-2]":goto2870
460  print"Heeft[SPACE]2[SPACE]gelijke[
     SPACE]wortels:[SPACE][CRSR-DOWN]"
470  print"[COM-7]x1[SPACE]=[SPACE]x2[S
     PACE]=[SPACE]";x"[CTRL-2]":goto287
     0
480  print"Is[SPACE]van[SPACE]de[SPACE]
     eerste[SPACE]graad[SPACE]met[SPACE
     ]als[SPACE]wortel:[SPACE][CRSR-DOW
     N]"
490  print"[COM-7]x[SPACE]=[SPACE]";x"[
     CTRL-2]":goto2870
```

```
500  print"[COM-7]Is[SPACE]een[SPACE]valse[SPACE]vergelijking[SPACE]!![CTRL 2]":goto2870
510  print"[COM-7]Is[SPACE]een[SPACE]identieke[SPACE]vergelijking[SPACE]!![CTRL-2]":goto2870
520  print"[SHIFT-CLR]"tab(9)"[CRSR-DOWN][COM-8]Oppervlakteberekening."
530  printtab(9)"---------------------[CRSR-DOWN]"
540  print"[CTRL-2][SPACE]1[SPACE]=[SPACE]Cirkel.":print"[CRSR-DOWN][SPACE]2[SPACE]=[SPACE]Bol.":print"[CRSR-DOWN][SPACE]3[SPACE]=[SPACE]Piramide[SPACE](met[SPACE]als";
550  print"[SPACE]grondvlak[SPACE]een[19xSPACE]vierkant)."
560  printtab(11)"[CRSR-DOWN][COM-8]Volumeberekening.":printtab(11)"-----------------[CRSR-DOWN]"
570  print"[CTRL-2][SPACE]4[SPACE]=[SPACE]Bol.":print"[CRSR-DOWN][SPACE]5[SPACE]=[SPACE]Piramide-Kegel.[CRSR-DOWN]"
580  print"[2xSPACE](Bij[SPACE]'5'[SPACE]wordt[SPACE]er[SPACE]rechtstreeks[SPACE]ge-"
590  print"[2xSPACE]bruik[SPACE]gemaakt[SPACE]v.d.[SPACE]oppervlakte[SPACE]van[6xSPACE]het[SPACE]grondvlak).[CRSR-UP]":gosub2730
600  getj$:ifj$=""then600
610  j=val(j$):on j goto 630,700,770,940,830
620  goto520
630  print"[SHIFT-CLR][CRSR-DOWN]Wat[SPACE]is[SPACE]er[SPACE]gevraagd[SPACE]?[SPACE](S[SPACE]of[SPACE]r)"
640  getj$:ifj$=""then640
650  ifj$=chr$(83)then680
660  ifj$=chr$(82)then690
670  goto630
680  print"[SHIFT-CLR][CRSR-DOWN]":gosub2770:s=*(r*r):goto2830
690  print"[SHIFT-CLR][CRSR-DOWN]":gosub2750:r=sqr(s/):goto2850
700  print"[SHIFT-CLR][CRSR-DOWN]Wat[SPACE]is[SPACE]er[SPACE]gevraagd[SPACE]?[SPACE](S[SPACE]of[SPACE]r)"
710  getj$:ifj$=""then710
720  ifj$=chr$(83)then750
730  ifj$=chr$(82)then760
740  goto700
750  print"[SHIFT-CLR][CRSR-DOWN]":gosub2770:s=4**(r*r):goto2830
760  print"[SHIFT-CLR][CRSR-DOWN]":gosub2750:r=sqr(s/(4*)):goto2850
770  print"[SHIFT-CLR][CRSR-DOWN]":gosub2740:input"Geef[SPACE]waarde[SPACE]van[SPACE]'z'[SPACE]";z:print
780  if(z*z)-((b/2)*(b/2))<=0thenprint"Foute[SPACE]cijfergegevens[SPACE]!!":goto820
790  x=sqr((z*z)-((b/2)*(b/2))):y=(x*b)/2:s=(y*4)+(b*b)
800  print"Opp.[SPACE]piramide[SPACE](4[SPACE]zijvlakken[SPACE]+[SPACE]grondvlak)":print"=[COM-7]"s"m[CRSR-UP]2[CRSR-DOWN][CTRL-2]"
810  print"[CRSR-DOWN](De[SPACE]4[SPACE]zijvlakken[SPACE]elk[SPACE]=[COM-7]"y"m[CRSR-UP]2[CRSR-DOWN][CTRL-2][SPACE])":goto2870
820  gosub2910:goto770
830  print"[SHIFT-CLR][CRSR-DOWN]Wat[SPACE]is[SPACE]er[SPACE]gevraagd[SPACE]?[SPACE](V,[SPACE]S[SPACE]of[SPACE]h)"
840  getj$:ifj$=""then840
850  ifj$=chr$(86)then890
860  ifj$=chr$(83)then900
870  ifj$=chr$(72)then920
880  goto830
890  print"[SHIFT-CLR][CRSR-DOWN]":gosub2750:gosub2760:v=(s*h)/3:goto2840
900  print"[SHIFT-CLR][CRSR-DOWN]":gosub2780:gosub2760:ifh=0then900
910  s=(3*v)/h:goto2830
920  print"[SHIFT-CLR][CRSR-DOWN]":gosub2780:gosub2750:ifs=0then920
930  h=(3*v)/s:print"[CRSR-DOWN]De[SPACE]hoogte[SPACE]=[COM-7]"h"m[CTRL-2]":goto2870
940  print"[SHIFT-CLR][CRSR-DOWN]Wat[SPACE]is[SPACE]er[SPACE]gevraagd[SPACE]?[SPACE](V[SPACE]of[SPACE]r)"
950  getj$:ifj$=""then950
960  ifj$=chr$(86)then990
970  ifj$=chr$(82)then1000
980  goto940
990  print"[SHIFT-CLR][CRSR-DOWN]":gosub2770:v=(4/3)**(r*r*r):goto2840
1000 print"[SHIFT-CLR][CRSR-DOWN]":gosub2780:r=(v/((4/3)*))~(1/3):goto2850
1010 print"[SHIFT-CLR][CRSR-DOWN][4xCRSR-RIGHT][COM-8]Procent-[SPACE]en[SPACE]Gemiddeldeberekening."
1020 print"[4xCRSR-RIGHT]-------------------------------[2xCRSR-DOWN]"
1030 print"[CTRL-2]f1)[SPACE]Procentberekening."
1040 print"[CRSR-DOWN]f3)[SPACE]Gemiddeldeberekening.":gosub2720
1050 getj$:ifj$=""then1050
1060 ifj$=chr$(133)then1090
1070 ifj$=chr$(134)then1130
1080 goto1010
1090 print"[SHIFT-CLR]"tab(11)"[CRSR-DOWN][COM-8]Procentberekening.":printtab(11)"------------------[2xCRSR-DOWN]"
1100 input"[CTRL-2]Maximum[SPACE]te[SPACE]behalen:[SPACE]";m:input"[CRSR-DOWN]Hoeveel[SPACE]is[SPACE]er[SPACE]behaald:[SPACE]";b
1110 print:ifm=0then1100
1120 t=m/10:p=(b/t)*10:printb "op"m"=[COM 7]"p"Procent[CTRL-2][SPACE]!":goto2870
1130 print"[SHIFT-CLR]"tab(9)"[CRSR-DOWN][COM-8]Gemiddeldeberekening."
1140 printtab(9)"---------------------[2xCRSR-DOWN]"
1150 print"[CTRL-2]Van[SPACE]hoeveel[SPACE]getallen[SPACE]wilt[SPACE]U[SP
```

```
     ACE]het"
1160 input"gemiddelde[SPACE]berekenen:[
     SPACE]";x:ifx=0then2870
1170 dimb(x):foru=1tox
1180 print"[CRSR-DOWN]Geef[SPACE]waarde
     [SPACE]van[SPACE]Nr.";u;:inputb(u)
     :next:so=0
1190 foru=1tox:so=so+b(u):next:print"[C
     RSR-DOWN]Het[SPACE]gemiddelde[SPAC
     E]=[COM-7]"som/x"[CTRL-2]!":goto28
     70
1200 print"[SHIFT-CLR]"tab(8)"[CRSR-DOW
     N][COM-8]Celsius[SPACE]<->[SPACE]F
     ahrenheit."
1210 printtab(8)"----------------------
     -[2xCRSR-DOWN]"
1220 print"[CTRL-2]f1)[SPACE]Celsius[SP
     ACE]->[SPACE]Fahrenheit."
1230 print"[CRSR-DOWN]f3)[SPACE]Fahrenh
     eit[SPACE]->[SPACE]Celsius.":gosub
     2720
1240 getj$:ifj$=""then1240
1250 ifj$=chr$(133)then1280
1260 ifj$=chr$(134)then1300
1270 goto1200
1280 input"[SHIFT-CLR][CRSR-DOWN]Geef[S
     PACE]het[SPACE]aantal[SPACE]graden
     [SPACE]Celsius:[SPACE]";c:print:f=
     (c*9)/5+32
1290 print""c"graden[SPACE]C.[SPACE]=[C
     OM 7]"f"graden[SPACE]F.[CTRL-2]":g
     oto2870
1300 input"[SHIFT-CLR][CRSR-DOWN]Geef[S
     PACE]het[SPACE]aantal[SPACE]graden
     [SPACE]Fahrenheit:[SPACE]";f:print
     :c=(f-32)*5/9
1310 print""f"graden[SPACE]F.[SPACE]=[C
     OM 7]"c"graden[SPACE]C.[CTRL-2]":g
     oto2870
1320 print"[SHIFT-CLR][CRSR-DOWN][6xCRS
     R-RIGHT][COM-8]Opbrengst[SPACE]van
     [SPACE]een[SPACE]kapitaal."
1330 print"[6xCRSR-RIGHT]--------------
     -------------[2xCRSR-DOWN]":print"
     [CTRL-2]Wat[SPACE]is[SPACE]er[SPAC
     E]gevraagd[SPACE]?"
1340 print"[20xCOM-T][CRSR-DOWN]":print
     "[CRSR-DOWN][SPACE]1[SPACE]=[SPACE
     ]Het[SPACE]eindkapitaal."
1350 print"[CRSR-DOWN][SPACE]2[SPACE]=[
     SPACE]Het[SPACE]beginkapitaal.":pr
     int"[CRSR-DOWN][SPACE]3[SPACE]=[SP
     ACE]De[SPACE]intrest."
1360 print"[CRSR-DOWN][SPACE]4[SPACE]=[
     SPACE]Het[SPACE]aantal[SPACE]jaar[
     SPACE]v.d.[SPACE]uitzetting.":gosu
     b2730
1370 getj$:ifj$=""then1370
1380 j=val(j$):on j goto 1400,1420,1440
     ,1460
1390 goto1320
1400 print"[SHIFT-CLR][CRSR-DOWN]":gosu
     b2790:gosub2800:gosub2810:g=k*(1+i
     /100)~n
1410 print"[CRSR-DOWN]Het[SPACE]eindkap
     itaal[SPACE]=[COM-7]"g"frank[CTRL-
     2][SPACE]!":goto2870
1420 print"[SHIFT-CLR][CRSR-DOWN]":gosu
     b2820:gosub2800:gosub2810:k=g/((1+
     i/100)~n)
1430 print"[CRSR-DOWN]Het[SPACE]beginka
     pitaal[SPACE]=[COM-7]"k"frank[CTRL
     2][SPACE]!":goto2870
1440 print"[SHIFT-CLR][CRSR-DOWN]":gosu
     b2790:gosub2820:gosub2810:ifk=0orn
     =0then1440
1450 i=100*(((g/k)~(1/n))-1):print"[CRS
     R-DOWN]De[SPACE]intrest[SPACE]=[CO
     M 7]"i"%[CTRL-2][SPACE]!":goto2870
1460 print"[SHIFT-CLR][CRSR-DOWN]":gosu
     b2790:gosub2820:gosub2800:ifk=0org
     =0ori=0then1460
1470 n=log(g/k)/log(1+i/100):print"[CRS
     R-DOWN]Het[SPACE]aantal[SPACE]jaar
     [SPACE]=[COM-7]"n"[CTRL-2]!":goto2
     870
1480 print"[SHIFT-CLR]"tab(10)"[CRSR-DO
     WN][COM-8]Wortels[SPACE]en[SPACE]m
     achten."
1490 printtab(10)"------------------[2
     xCRSR-DOWN]":print"[CTRL-2]f1)[SPA
     CE]Worteltrekkingen."
1500 print"[CRSR-DOWN]f3)[SPACE]Machten
     .":gosub2720
1510 getj$:ifj$=""then1510
1520 ifj$=chr$(133)then1550
1530 ifj$=chr$(134)then1590
1540 goto1480
1550 print"[SHIFT-CLR]"tab(11)"[CRSR-DO
     WN][COM-8]Worteltrekkingen.":print
     tab(11)"-----------------[2xCRSR-D
     OWN]"
1560 input"[CTRL-2]Geef[SPACE]waarde[SP
     ACE]v.d.[SPACE]machtswortel:[SPACE
     ]";m:ifm=0then1550
1570 input"[CRSR-DOWN]Uit[SPACE]welk[SP
     ACE]getal:[SPACE]";g:w=g~(1/m)
1580 print"[CRSR-DOWN]De"m"-de[SPACE]ma
     chtswortel[SPACE]uit"g"":print"[CR
     SR-DOWN]=[COM-7]"w"[CTRL-2]!":goto
     2870
1590 print"[SHIFT-CLR]"tab(16)"[CRSR-DO
     WN][COM-8]Machten.":printtab(16)"-
     -------[2xCRSR-DOWN]"
1600 input"[CTRL-2]Geef[SPACE]waarde[SP
     ACE]v.d.[SPACE]macht:[SPACE]";m:in
     put"[CRSR-DOWN]Van[SPACE]welk[SPAC
     E]getal:[SPACE]";g:o=g~m
1610 print"[CRSR-DOWN]De"m"-de[SPACE]ma
     cht[SPACE]van"g"=[COM-7]"o"[CTRL-2
     ]!":goto2870
1620 print"[SHIFT-CLR]"tab(6)"[CRSR-DOW
     N][COM-8]Sinus,[SPACE]Cosinus,[SPA
     CE]Tangens,..."
1630 printtab(6)"----------------------
     -----[2xCRSR-DOWN]"
1640 print"[CTRL-2]f1)[SPACE]Hoek[SPACE
     ]->[SPACE]Goniometrische[SPACE]get
     allen."
1650 print"[CRSR-DOWN]f3)[SPACE]Goniome
     trisch[SPACE]getal[SPACE]->[SPACE]
     Hoek.":gosub2720
1660 getj$:ifj$=""then1660
1670 ifj$=chr$(133)then1700
1680 ifj$=chr$(134)then1820
1690 goto1620
```

```
1700 print"[SHIFT-CLR]"tab(4)"[CRSR-DOW
     N][COM-8]Hoek[SPACE]->[SPACE]Gonio
     metrische[SPACE]getallen."
1710 printtab(4)"----------------------
     ----------[2xCRSR-DOWN]"
1720 print"[CTRL-2]Geef[SPACE]de[SPACE]
     waarde[SPACE]van[SPACE]de[SPACE]ho
     ek[SPACE](in[SPACE]graden):":input
     h
1730 ifh=0orh=360then1990
1740 ifh=180then2000
1750 ifh=90then2010
1760 si=sin(h/57.29578):si$=str$(si):co
     =cos(h/57.29578):co$=str$(co)
1770 tg=tan(h/57.29578):tg$=str$(tg):cs
     =1/sin(h/57.29578):cs$=str$(cs)
1780 se=1/cos(h/57.29578):se$=str$(se):
     ct=1/tan(h/57.29578):ct$=str$(ct)
1790 print"[CRSR-DOWN]SINUS[5xSPACE]=[C
     OM 7]"si$"[CTRL-2]":print"COSINUS[
     3xSPACE]=[COM-7]"co$"[CTRL-2]"
1800 print"TANGENS[3xSPACE]=[COM-7]"tg$
     "[CTRL-2]":print"COSECANS[2xSPACE]
     =[COM-7]"cs$"[CTRL-2]"
1810 print"SECANS[4xSPACE]=[COM-7]"se$"
     [CTRL-2]":print"COTANGENS[SPACE]=[
     COM-7]"ct$"[CTRL-2]":goto2870
1820 print"[SHIFT-CLR]"tab(6)"[CRSR-DOW
     N][COM-8]Goniometrisch[SPACE]getal
     [SPACE]->[SPACE]Hoek."
1830 printtab(6)"----------------------
     ------[2xCRSR-DOWN]":print"[CTRL-2
     ]Wat[SPACE]is[SPACE]er[SPACE]gegev
     en[SPACE]?"
1840 print"[CRSR-DOWN][SPACE]1[SPACE]=[
     SPACE]Sinus":print"[CRSR-DOWN][SPA
     CE]2[SPACE]=[SPACE]Cosinus":print"
     [CRSR-DOWN][SPACE]3[SPACE]=[SPACE]
     Tangens":gosub2730
1850 getj$:ifj$=""then1850
1860 j=val(j$):on j goto 1880,1920,1960
1870 goto1820
1880 input"[SHIFT-CLR][CRSR-DOWN]Welk[S
     PACE]is[SPACE]de[SPACE]sinus[SPACE
     ]";si:ifsi>1orsi<-1then1970
1890 ifsi=1thenh=90:goto2860
1900 ifsi=-1thenh=-90:goto2860
1910 h=atn(si/sqr(-si*si+1))*57.29578:g
     oto2860
1920 input"[SHIFT-CLR][CRSR-DOWN]Welk[S
     PACE]is[SPACE]de[SPACE]cosinus[SPA
     CE]";co:ifco>1orco<-1then1980
1930 ifco=1thenh=0:goto2860
1940 ifco=-1thenh=180:goto2860
1950 h=(-atn(co/sqr(-co*co+1))+/2)*57.2
     9578:goto2860
1960 input"[SHIFT-CLR][CRSR-DOWN]Welk[S
     PACE]is[SPACE]de[SPACE]tangens[SPA
     CE]";tg:h=atn(tg)*57.29578:goto286
     0
1970 print"[CRSR-DOWN]Foutieve[SPACE]wa
     arde[SPACE]!![CRSR-DOWN]":gosub291
     0:goto1880
1980 print"[CRSR-DOWN]Foutieve[SPACE]wa
     arde[SPACE]!![CRSR-DOWN]":gosub291
     0:goto1920
1990 si$="[SPACE]0":co$="[SPACE]1":tg$=
     "[SPACE]0":cs$="[SPACE]/":se$="[SP
     ACE]1":ct$="[SPACE]/":goto1790
2000 si$="[SPACE]0":co$="-1":tg$="[SPAC
     E]0":cs$="[SPACE]/":se$="-1":ct$="
     [SPACE]/":goto1790
2010 si$="[SPACE]1":co$="[SPACE]0":tg$=
     "[SPACE]/":cs$="[SPACE]1":se$="[SP
     ACE]/":ct$="[SPACE]0":goto1790
2020 print"[SHIFT-CLR][CRSR-DOWN][COM-8
     ][2xSPACE]Weekdagen[SPACE]bepalen[
     SPACE]en[SPACE]priemgetallen."
2030 print"[2xSPACE]-------------------
     ----------------[CTRL-2][CRSR-DOWN
     ]"
2040 print"f1)[SPACE]Weekdagen[SPACE]be
     palen.":print"[CRSR-DOWN]f3)[SPACE
     ]Priemgetal[SPACE]of[SPACE]niet[SP
     ACE]?!"
2050 print"[CRSR-DOWN]f5)[SPACE]Alle[SP
     ACE]priemgetallen[SPACE]tot[SPACE]
     ...":gosub2720
2060 getj$:ifj$=""then2060
2070 ifj$=chr$(133)then2400
2080 ifj$=chr$(134)then2260
2090 ifj$=chr$(135)then2110
2100 goto2020
2110 print"[SHIFT-CLR]"tab(7)"[CRSR-DOW
     N][COM-8]Alle[SPACE]priemgetallen[
     SPACE]tot[SPACE]..."
2120 printtab(7)"----------------------
     ----[CTRL-2][CRSR-DOWN]"
2130 input"Alle[SPACE]priemgetallen[SPA
     CE]t/m[SPACE]welk[SPACE]getal[SPAC
     E]?[5xSPACE]";p:print:z=0:g=4:a=1
2140 gosub2200
2150 ifa>pthen2190
2160 z=z+1
2170 print"[COM-7][CRSR-RIGHT]"a"[CTRL-
     2]";:ifint(z/5)=z/5thenprint
2180 goto2140
2190 print"[CRSR-UP][CRSR-DOWN]":goto28
     70
2200 a=a+2:d=3:a1=sqr(a)
2210 ifd>a1thenreturn
2220 c=a/d
2230 ifc=int(c)then2200
2240 d=d+2
2250 goto2210
2260 print"[SHIFT-CLR]"tab(9)"[CRSR-DOW
     N][COM-8]Priemgetal[SPACE]of[SPACE
     ]niet[SPACE]?!"
2270 printtab(9)"---------------------[
     CTRL-2][CRSR-DOWN]"
2280 print"Voer[SPACE]het[SPACE]getal[S
     PACE]in[SPACE]dat[SPACE]moet[SPACE
     ]onderzocht"
2290 input"worden[SPACE](groter[SPACE]d
     an[SPACE]1)[SPACE]";g:print"[CRSR-
     DOWN][CRSR-RIGHT][CTRL-9]EVEN[SPAC
     E]GEDULD[SPACE]A.U.B."
2300 ifg<=1org<>int(g)then2260
2310 ifg<4then2390
2320 l=sqr(g):d=2
2330 q=g/d
2340 ifq<>int(q)then2370
2350 foru=1305to1322:pokeu,32:next:prin
     t"[CRSR-DOWN]"g"=[SPACE][COM-7][CT
     RL 9]GEEN[SPACE]priemgetal[SPACE]!
     ![CTRL-2]"
```

```
2360 print"[CRSR-DOWN][SPACE]De[SPACE]k
     leinste[SPACE]deler[SPACE]is[COM-7
     ]"d"[CTRL-2]":goto2870
2370 d=d+1:ifd>1then2390
2380 goto2330
2390 foru=1305to1322:pokeu,32:next:prin
     t"[CRSR-DOWN]"g"=[SPACE][COM-7][CT
     RL 9]PRIEMGETAL[SPACE]!![CTRL-2]":
     goto2870
2400 print"[SHIFT-CLR]"tab(11)"[CRSR-DO
     WN][COM-8]Weekdagen[SPACE]bepalen.
     ":printtab(11)"-----------------[
     CTRL-2]"
2410 dimdw$(8),mi(13)
2420 dw$(0)="ZATERDAG":dw$(1)="ZONDAG":
     dw$(2)="MAANDAG":dw$(3)="DINSDAG"
2430 dw$(4)="WOENSDAG":dw$(5)="DONDERDA
     G":dw$(6)="VRIJDAG"
2440 mi(1)=1:mi(2)=4:mi(3)=4:mi(4)=0:mi
     (5)=2:mi(6)=5:mi(7)=0:mi(8)=3:mi(9
     )=6
2450 mi(10)=8:mi(11)=4:mi(12)=6
2460 print"[CRSR-DOWN]Geef[SPACE]de[SPA
     CE]dag,[SPACE]de[SPACE]maand[SPACE
     ]en[SPACE]het[SPACE]jaar,"
2470 input"gescheiden[SPACE]door[SPACE]
     komma's[SPACE](dd,mm,jj).[5xSPACE]
     ";da,mo,yr
2480 ifmo<1ormo>12then2560
2490 ifda<1orda>31then2560
2500 ifyr<0oryr>99then2560
2510 gosub2570
2520 wo=(int(yr/4))+yr:wo=wo+mi(mo)+da:
     ws=wo:wo=int(wo/7):wo=ws-(wo*7)
2530 ifle=1thenwo=wo-1
2540 ifwo<0thenwo=wo+7
2550 print"[2xCRSR-DOWN]"da;"/";mo;"/";
     yr;"[SPACE]=[SPACE][COM-7]";dw$(wo
     ):print"[CTRL-2]":goto2870
2560 print"[2xCRSR-DOWN]De[SPACE]ingevo
     erde[SPACE]datum[SPACE]is[SPACE]on
     juist[SPACE]!!":goto2870
2570 ifmo=4andda>30then2560
2580 ifmo=6andda>30then2560
2590 ifmo=9andda>30then2560
2600 ifmo=11andda>30then2560
2610 ifint(yr/4)=yr/4thenle=1
2620 ifmo>2thenle=0
2630 ifmo<>2thenreturn
2640 ifda<29thenreturn
2650 ifda>29then2560
2660 wy=int(yr/4):wy=yr-(wy*4)
2670 ifwy<>0then2560
2680 return
2690 print"[SHIFT-CLR]":poke53280,14
2700 print"[COM-7][4xSPACE]****[SPACE]c
     ommodore[SPACE]64[SPACE]basic[SPAC
     E]v2[SPACE]****[CRSR-DOWN]"chr$(14
     2)
2710 print"[SPACE]64k[SPACE]ram[SPACE]s
     ystem[2xSPACE]38911[SPACE]basic[SP
     ACE]bytes[SPACE]free":poke788,49:p
     oke657,0:new
2720 print"[2xCRSR-DOWN][CTRL-9]Druk[SP
     ACE]de[SPACE]functietoets[SPACE]va
     n[SPACE]uw[SPACE]keuze[SPACE]!":re
     turn
2730 print"[2xCRSR-DOWN][CTRL-9]Druk[SP
     ACE]de[SPACE]cijfer-toets[SPACE]va
     n[SPACE]uw[SPACE]keuze[SPACE]!":re
     turn
2740 input"Geef[SPACE]waarde[SPACE]van[
     SPACE]'b'[SPACE]";b:print:return
2750 input"Geef[SPACE]waarde[SPACE]van[
     SPACE]'S'[SPACE]";s:print:return
2760 input"Geef[SPACE]waarde[SPACE]van[
     SPACE]'h'[SPACE]";h:print:return
2770 input"Geef[SPACE]waarde[SPACE]van[
     SPACE]'r'[SPACE]";r:print:return
2780 input"Geef[SPACE]waarde[SPACE]van[
     SPACE]'V'[SPACE]";v:print:return
2790 print"Geef[SPACE]waarde[SPACE]van[
     SPACE]beginkapitaal:[SPACE]":input
     k:print:return
2800 print"Geef[SPACE]waarde[SPACE]van[
     SPACE]de[SPACE]intrest[SPACE](in[S
     PACE]%):[SPACE]":inputi:print:retu
     rn
2810 print"Geef[SPACE]waarde[SPACE]van[
     SPACE]het[SPACE]aantal[SPACE]uitge
     zette[3xSPACE]jaren:[SPACE]":input
     n:print:return
2820 print"Geef[SPACE]waarde[SPACE]van[
     SPACE]eindkapitaal:[SPACE]":inputg
     :print:return
2830 print"[CRSR-DOWN]De[SPACE]oppervla
     kte[SPACE]=[COM-7]"s"m[CRSR-UP]2[C
     RSR-DOWN][CTRL-2]":goto2870
2840 print"[CRSR-DOWN]Het[SPACE]volume[
     SPACE]=[COM-7]"v"m[CRSR-UP]3[CRSR-
     DOWN][CTRL-2]":goto2870
2850 print"[CRSR-DOWN]De[SPACE]straal[S
     PACE]=[COM-7]"r"m[CTRL-2]":goto287
     0
2860 print"[CRSR-DOWN]De[SPACE]hoek[SPA
     CE]=[COM-7]"h"graden[CTRL-2][SPACE
     ]!"
2870 print"[2xCRSR-DOWN][CTRL-9]Druk[SP
     ACE]op[SPACE]'RETURN'[SPACE]om[SPA
     CE]terug[SPACE]te[SPACE]keren":pri
     nt"[CTRL-9]naar[SPACE]het[SPACE]me
     nu[SPACE]!!"
2880 getj$:ifj$=""then2880
2890 ifj$=chr$(13)then140
2900 goto2870
2910 foru=1to999:next:return
2920 end
```

** EINDE LISTING rekenwonder **

Checksum Reken

REGEL 1	89
REGEL 2	123
REGEL 3	89
REGEL 4	62
REGEL 5	81
REGEL 6	45
REGEL 7	168
REGEL 8	55
REGEL 9	81
REGEL 10	121
REGEL 20	43
REGEL 30	67
REGEL 40	176
REGEL 50	180
REGEL 60	237
REGEL 70	217
REGEL 80	237
REGEL 90	166
REGEL 100	167
REGEL 110	82
REGEL 120	73
REGEL 130	83
REGEL 140	91
REGEL 150	202
REGEL 160	133
REGEL 170	42
REGEL 180	127
REGEL 190	210
REGEL 200	182
REGEL 210	222
REGEL 220	170
REGEL 230	116
REGEL 240	39
REGEL 250	209
REGEL 260	30
REGEL 270	125
REGEL 280	25
REGEL 290	150
REGEL 300	33
REGEL 310	120
REGEL 320	195
REGEL 330	187
REGEL 340	13
REGEL 350	239
REGEL 360	181
REGEL 370	113
REGEL 380	234
REGEL 390	70
REGEL 400	237
REGEL 410	30
REGEL 420	97
REGEL 430	77
REGEL 440	36
REGEL 450	146
REGEL 460	113
REGEL 470	116
REGEL 480	124
REGEL 490	124
REGEL 500	66
REGEL 510	89
REGEL 520	58
REGEL 530	209
REGEL 540	102
REGEL 550	24
REGEL 560	106
REGEL 570	151
REGEL 580	84
REGEL 590	128
REGEL 600	117
REGEL 610	212
REGEL 620	32
REGEL 630	171
REGEL 640	121
REGEL 650	115
REGEL 660	115
REGEL 670	34
REGEL 680	51
REGEL 690	240
REGEL 700	171
REGEL 710	119
REGEL 720	113
REGEL 730	113
REGEL 740	32
REGEL 750	19
REGEL 760	33
REGEL 770	88
REGEL 780	138
REGEL 790	1
REGEL 800	193
REGEL 810	127
REGEL 820	186
REGEL 830	163
REGEL 840	123
REGEL 850	121
REGEL 860	110
REGEL 870	110
REGEL 880	36
REGEL 890	247
REGEL 900	222
REGEL 910	16
REGEL 920	234
REGEL 930	192
REGEL 940	174
REGEL 950	125
REGEL 960	122
REGEL 970	149
REGEL 980	38
REGEL 990	70
REGEL 1000	174
REGEL 1010	55
REGEL 1020	64
REGEL 1030	37
REGEL 1040	110
REGEL 1050	165
REGEL 1060	203
REGEL 1070	199
REGEL 1080	75
REGEL 1090	194
REGEL 1100	29
REGEL 1110	246
REGEL 1120	135
REGEL 1130	177
REGEL 1140	181
REGEL 1150	145
REGEL 1160	96
REGEL 1170	96
REGEL 1180	88
REGEL 1190	73
REGEL 1200	47
REGEL 1210	14
REGEL 1220	17
REGEL 1230	177
REGEL 1240	166
REGEL 1250	204
REGEL 1260	198
REGEL 1270	76
REGEL 1280	6
REGEL 1290	165
REGEL 1300	208
REGEL 1310	165
REGEL 1320	48
REGEL 1330	167
REGEL 1340	54
REGEL 1350	144
REGEL 1360	93
REGEL 1370	170
REGEL 1380	194
REGEL 1390	79
REGEL 1400	22
REGEL 1410	68
REGEL 1420	98
REGEL 1430	141
REGEL 1440	129
REGEL 1450	14
REGEL 1460	86
REGEL 1470	227
REGEL 1480	254
REGEL 1490	188
REGEL 1500	208
REGEL 1510	166
REGEL 1520	204
REGEL 1530	209
REGEL 1540	86
REGEL 1550	111
REGEL 1560	212
REGEL 1570	92
REGEL 1580	172
REGEL 1590	21
REGEL 1600	144
REGEL 1610	63
REGEL 1620	182
REGEL 1630	192
REGEL 1640	174
REGEL 1650	42
REGEL 1660	172
REGEL 1670	201
REGEL 1680	205
REGEL 1690	82
REGEL 1700	140
REGEL 1710	159
REGEL 1720	37
REGEL 1730	114
REGEL 1740	135
REGEL 1750	88
REGEL 1760	205
REGEL 1770	182
REGEL 1780	141
REGEL 1790	113
REGEL 1800	140
REGEL 1810	30
REGEL 1820	106
REGEL 1830	232
REGEL 1840	100
REGEL 1850	173
REGEL 1860	227
REGEL 1870	84
REGEL 1880	68
REGEL 1890	167
REGEL 1900	253
REGEL 1910	178
REGEL 1920	185
REGEL 1930	100
REGEL 1940	120
REGEL 1950	24
REGEL 1960	244
REGEL 1970	3
REGEL 1980	254
REGEL 1990	0
REGEL 2000	90
REGEL 2010	0
REGEL 2020	168
REGEL 2030	26
REGEL 2040	6
REGEL 2050	12
REGEL 2060	167
REGEL 2070	199
REGEL 2080	204
REGEL 2090	199
REGEL 2100	77
REGEL 2110	75
REGEL 2120	136
REGEL 2130	120
REGEL 2140	81
REGEL 2150	64
REGEL 2160	65
REGEL 2170	142
REGEL 2180	80
REGEL 2190	19
REGEL 2200	29
REGEL 2210	39
REGEL 2220	39
REGEL 2230	52
REGEL 2240	22
REGEL 2250	78
REGEL 2260	244
REGEL 2270	169
REGEL 2280	140
REGEL 2290	94
REGEL 2300	129
REGEL 2310	46
REGEL 2320	178
REGEL 2330	59
REGEL 2340	10
REGEL 2350	28
REGEL 2360	25
REGEL 2370	144
REGEL 2380	81
REGEL 2390	145
REGEL 2400	50
REGEL 2410	69
REGEL 2420	132
REGEL 2430	12
REGEL 2440	215
REGEL 2450	7
REGEL 2460	174
REGEL 2470	167
REGEL 2480	223
REGEL 2490	178
REGEL 2500	11
REGEL 2510	91
REGEL 2520	67
REGEL 2530	128
REGEL 2540	154
REGEL 2550	148
REGEL 2560	9
REGEL 2570	201
REGEL 2580	203
REGEL 2590	206
REGEL 2600	247
REGEL 2610	118
REGEL 2620	36
REGEL 2630	242
REGEL 2640	99
REGEL 2650	160
REGEL 2660	199
REGEL 2670	67
REGEL 2680	142
REGEL 2690	212
REGEL 2700	128
REGEL 2710	36
REGEL 2720	235
REGEL 2730	189
REGEL 2740	161
REGEL 2750	67
REGEL 2760	173
REGEL 2770	193
REGEL 2780	73
REGEL 2790	152
REGEL 2800	169
REGEL 2810	251
REGEL 2820	79
REGEL 2830	108
REGEL 2840	83
REGEL 2850	17
REGEL 2860	88
REGEL 2870	104
REGEL 2880	177
REGEL 2890	99
REGEL 2900	90
REGEL 2910	140
REGEL 2920	128

Filerequester in AmigaBasic

De volgende Amiga-programma's zijn van de hand van Ronald Sloot. Voor de regelmatige lezers van ons blad is dit zeker geen onbekende. Hij stuurde ons weer een aantal routines die de AmigaBasic gebruikers het leven een stuk makkelijker maken (voor zover dit mogelijk is). De routines zijn goed becommentarieerd, zodat gebruikers de routines naar eigen smaak kunnen aanpassen. Dit komt natuurlijk de duidelijkheid ten goede. We nemen hier ook de volledige uitleg op die hij bij zijn programma's heeft geschreven.

Informatie over de routines FileRequest, Invoer v2.0, Match en Request .
Goede FileRequesters, invoerroutines die werken met scrollbare tekst en pattern matching, ik heb ze nog nooit (of bijna nooit) in AmigaBASIC gezien... Daar hoop ik met deze routines verandering in te brengen.

De behandeling van Match

Met match kunnen voorwaarden aan een tekst gesteld worden. Voorbeelden zouden kunnen zijn: de naam moet eindigen op Janssen, de tekst moet beginnen met JOY en daarachter mag een cijfer tussen de 1 en de 9 staan, de tekst moet de tekst DMA bevatten en eindigen op .h. De voorwaarden aan de tekst worden met bepaalde tekens gezet. Deze routine gebruikt een asteriks (*) een vraagteken (?) en een begin-en eindhaak ([]). Deze tekens hebben elk een bepaalde betekenis. Zo is de betekenis van "*" een string

```
Workbench Screen
ASDG-RAM
R.Sloot

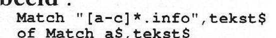
MOD(AUTOR & A).RUN
.info
MOD(R_I_M).RUN
(DIR) c
Lees Mij Eerst.info
(DIR) l
MOD(Req Inv Mat)
(DIR) DEVS
AmigaBASIC
(DIR) s

DF0: | DF1: | DH0: | RAM: | VD0: | *[]? | /
Dir  DF0:
File MOD(R_I_M).RUN
[ In orde ]  [ Afbreken ]
```

(tekst) met een willekeurige lengte en kan * dus inhouden aa azeezb zd 3342ere34 of helemaal geen teken. De betekenis van het "?" is een willekeurig teken welke bv. kan betekenen een a,d,z,1,3,4 of welk teken dan ook, maar houdt wel in de gaten dat het 1 teken is! (geen 0 of meer dan 1). Verder is de betekenis van de haken "[]" een bereik waar een bepaald teken aan moet voldoen. Bereiken zijn bijvoorbeeld [1-4] (het teken moet voldoen aan 1,2,3 of 4), [aert] (het teken moet voldoen aan a,e,r of een t), [aceg-k] (het teken moet voldoen aan een a,c,e of liggen binnen g-k (ghijk)) en als laatste [1a-z] (het teken moet voldoen aan een 1 of een letter zijn). Let wel, deze routine maakt geen verschil tussen hoofd- of kleine letters. (NIET CASE-SENSITIVE). Dit is wel heel makkelijk te verwezenlijken, maar dat wordt als commentaar bij de match-routine uitgelegd.

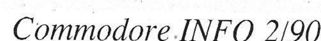
Hoe gebruikt men Match nu in AmigaBasic?

Nadat u de Match-routine via bijvoorbeeld COPY en PASTE in uw eigen programmalisting heeft geplaatst kunt u er gebruik van maken. U kunt dan de routine aanroepen met Match pattern$,vergelijk$. Pattern$ bevat dan de string met de voorwaarden (met de tekens *,? en [] dus) en vergelijk$ bevat dan de string die vergeleken dient te worden.
Bijvoorbeeld :



```
Match "[a-c]*.info",tekst$
of Match a$,tekst$
```

```
Invoer 10, 100, 60, 200, 1, 2, 0, 1, "Al ingevoerd:
        !    !    !    !   !  !  !  !      !
        /    !    !    !   \  !  Type \   tekst die al in box geplaatst wordt
max aantal   !    ! breedte \! Invoer \
tekens       !    ! v.d.box  \          Actie op return
             !    !           voorgrond,
          xpos ypos           achtergrondkleur
          (in pixels)
```



Voorbeelden pattern strings:

° *.info ;tekst moet eindigen op .info;

° *.info* ;er moet ".info" in de tekst zitten;

° *A*&* ;er moet een A en een & in de tekst zitten (de & moet dan wel na de A staan);

° ? ;tekst moet 1 teken bevatten;

° ?? ;tekst moet 2 tekens bevatten;

° a?? ;tekst moet beginnen met a en daarna moeten er 2 karakters staan (willekeurige karakters);

° [1-4] ;tekst bevat 1 teken dat moet bestaan uit een 1,2,3 of 4;

° *[ao-q] ;tekst moet eindigen op een a,o,p of q;

° [a-y]??*.PC ;tekst moet beginnen met teken dat ligt tussen a en y (incl. a en y) en daarna moeten 2 willekeurige karakters staan en de tekst moet eindigen op .PC (de minimale lengte is dus 1+1+ 1+3=6).

De behandeling van Invoer v2.0

Invoer v2.0 is bedoeld als vervanging van de input-routine van AmigaBasic zelf maar kan ook met enige moeite omgezet worden voor GFA-BASIC. De routine ondersteunt de cursortoetsen, DEL en Backspacetoets. Verder heeft hij nog wat extraatjes, welke zijn: volgend en vorig woord via de F9 (vorig) en F10 (volgend) toetsen maar ook springen naar begin en einde van de tekst (Cursor Omhoog/Cursor Omlaag), het wissen van de tekst door middel van de ESC-toets en de mogelijkheid om alleen bepaalde toetsen toe te staan bij de invoer zoals alleen de jnJNyY toetsen. De jnJNyY is er 1 van de 5 : 0 = alle tekens

1 = 012456789.-.*/^()+=

2 = cijfers + letters

3 = jnJNyY

4 = 0123456789.

Overigens : een eigen type is met enig inzicht in de routine heel makkelijk bij te voegen. Dan is er nog de actie op return. U kunt de routine meedelen (via een parameter van Invoer) of het toegestaan is geen tekst in te voeren (de tekst is leeg :""). Dit kan soms wel eens handig zijn als het echt nodig is dat er tekst ingevoerd moet zijn. Ook kunt u als parameter een tekst aan Invoer meegeven die hij dan alvast als invoer moet beschouwen (er staat bijvoorbeeld al gelijk "DF0:" in).
Nadat via bijvoorbeeld COPY & PASTE de routine in uw eigen programma gezet is, kan de routine gebruikt worden. Een voorbeeld daarvan staat in figuur 1.

De behandeling van FileRequest

FileRequest is een routine waarmee u vriendelijk een file kunt selecteren om deze te laden, saven, printen etcetera; u bedenkt het maar. Wanneer het FileRequest window wordt geopend, wordt er eerst gevraagd een device aan te klikken (of u kunt hem in de dirbox intikken). Van deze manier ben ik een voorstander (de andere manier is om gelijk de directory te gaan laden, bijvoorbeeld DF0: of de directory van de vorige keer), omdat je in 80% van de gevallen een andere directory wilt zien dan de standaard instelling van het programma. Ik moet hierbij opmerken dat deze filerequester zijn files van de vorige keer nog onthoudt, en dat hij dan niet vraagt om een device te kiezen, maar ook niet begint met laden, zodat het in principe hetzelfde is (en nog vriendelijk ook). Wat wel belangrijk is om op te merken, is dat deze routine gebruik maakt van de routines Invoer v2.0 en Match (v1.0) die hierboven beschreven zijn. Dus als deze FileRequest-routine gebruikt wordt, moeten Invoer v2.0 en Match ook een deel uitmaken van het programma. De filerequester is ook van het type "KLIK MAAR RAAK" i.p.v. "ER ZIJN NOG 20 WACHTENDEN VOOR U". Het "klik maar raak"-principe (zo heb ik het maar genoemd) houdt in dat tijdens het laden van de directory u een andere directory kunt kiezen, naar de vorige directory kunt gaan, een file kunt kiezen en kunt veranderen van device (en op In Orde en Afbreken kunt drukken) oftewel op alles volgt direct een actie. Een mogelijkheid van deze filerequester die u bijna nooit vindt bij andere filerequesters (en al is dat wel zo, dan is het meestal in de meest simpele vorm die er is, waarmee ik het achtervoegsel controle bedoel), is dat aan de entries van de directory voorwaarden gesteld kunnen worden zoals bijvoorbeeld: Kick*v1.[1-4]. Voor de uitleg van het pattern matchen verwijs ik terug naar de uitleg van de Match-routine in deze file.

De behandeling van Request

Request is een routine waarmee u zelf AutoRequesters kunt maken. U kent ze wel van "Insert Naam_Disk in any drive". Ik dacht even weer op een Commodore 64 te zitten met al dat gepoke.. De AC-Basic-compiler kwam trouwens bij deze routine in de problemen (v1.3) wanneer er een adres van een string werd opgevraagd die als data ingelezen was. Daar ben ik wel even een tijdje mee aan het klungelen geweest. Request verwacht dat de datapointer naar data staat die specifiek voor deze routine is. In het commentaar in het programma wordt de verwachte syntax uitgelegd, eveneens het wijzigen van de datapointer (RESTORE-functie). Verdere informatie staat in het programma...

Loader

```
' Loader van programmafile
' met de modules:
' - FileRequest
' - Invoer v2.0
' - Match  v1.0
CLEAR ,32000
LOAD "MOD(FileReq).MAIN",r
```

Filereq

```
CLEAR ,40000&

DECLARE FUNCTION ExNext& LIBRARY
DECLARE FUNCTION Examine& LIBRARY
DECLARE FUNCTION AllocMem& LIBRARY
DECLARE FUNCTION Move& LIBRARY
DECLARE FUNCTION Lock& LIBRARY

' CInfo / Datum 22 Oktober 1989
' Modules FileRequest / Invoer v2.0 / Match v1.0
'==============================================
' Voor eventuele problemen: R.Sloot / 01659-3686
' Haal A.U.B. voor uw eigen versie het kommentaar
' uit de listing zodat het (een beetje) sneller
' kan lopen....

libpath$="VD0:"
LIBRARY libpath$+"graphics.library"
LIBRARY libpath$+"dos.library"
LIBRARY libpath$+"exec.library"

 GOSUB InitFileReq
 GOSUB Demo

 END

InitFileReq:
'====Initialiseer/dimensioneer arrays benodigd voor subroutine
FileRequest

 MAXFILES% = 90
 GADGETS%  = 13
 DEVGADGH% = 7

 RESTORE FReqData
 DIM file$(MAXFILES%), Device$(DEVGADGH%), Kleur%(MAXFILES%),
gX%(GADGETS%), gY%(GADGETS%), gX1%(GADGETS%), gY1%(GADGETS%)
 DIM SHARED CursorBeeld%(30)

 'lees arrays in
 FOR i%=0 TO DEVGADGH%-1:READ Device$(i%):NEXT
 FOR i%=0 TO GADGETS%-1:READ gX%(i%), gY%(i%), gX1%(i%), gY1%(i%):NEXT

RETURN

Demo:
 CALL SetDrMd(WINDOW(8),0)
```

```
 COLOR 3
 Prin 5,10, "FileRequester"
 Prin 5,17, "============="
 COLOR 1
 Prin 5,26, "Syntax: FileRequest
x_window,y_window,boodschaptekst,schermnummer"
 Prin 20,40,"*voor de aanroep van deze routine moet deze
eerst geinitialiseerd"
 Prin 20,48," worden met GOSUB InitFileReq."
 Prin 20,58,"*de routine gebruikt de InvoerV2.0 en MatchV1.0
routines"
 COLOR 3
 Prin 5,70, "Invoer v2.0"
 Prin 5,77, "==========="
 COLOR 1
 Prin 5,86, "Syntax: Invoer
aantal_tekens,x_pos,y_pos,breedte_v.d_box,voorgrondkleur,"
 Prin 5,94,"
achtergrondkleur,invoertype,aktie_op_returntoets,tekst"
 Prin 5,106," * Invoer v2.0 is een BOXinvoerroutine
(scrollbaar)."
 Prin 5,114,"   Extra's: vorig en volgend woord, begin en
eind tekst en wis heel de tekst"
 COLOR 3
 Prin 5,136,"Match v1.0"
 Prin 5,144,"=========="
 COLOR 1
 Prin 5,153,"Syntax: Match patternstring,vergelijkingsstring"
 Prin 5,165," * Bereik :[], string met variabele lengte :*,
variabel teken :?"
 CALL SetDrMd&(WINDOW(8),1)

 COLOR 3:Prin 170,184,"< Druk op de linker muistoets >"
 WHILE MOUSE(0)=0:WEND

 '-1 is het workbenchscreen
 FileRequest 250,50,"Laad file",-1

 '===2e maal aanroepen om te laten zien dat entrys onthouden
blijven
 FileRequest 250,50,"Opslaan file",-1

 COLOR 1:CLS
 GOSUB DemoPatternMatching
RETURN

DemoPatternMatching:
 Prin 440,10,"Demo Pattern Matching"
 RESTORE PMdemoData
 DIM vergelijk$(12)

 FOR i%=0 TO 11:READ vergelijk$(i%):NEXT
 FOR i%=0 TO 8:READ testpat$(i%):NEXT

 COLOR 2:CALL RectFill&(WINDOW(8),20,20,190,150)
 LINE (20,20)-STEP(170,130),1,b
 COLOR 3:CALL RectFill&(WINDOW(8),230,20,395,120)
 LINE (230,20)-STEP(165,100),1,b
 COLOR 1
 Prin 20,16,"Vergelijkingstekst"
 Prin 230,16,"Testpatterns"
 COLOR 1,2:FOR i%=0 TO 11:Prin
25,30+i%*10,vergelijk$(i%):NEXT
 COLOR 1,3:FOR i%=0 TO 8:Prin 235,30+i%*10,testpat$(i%):NEXT

 testpat%=0
 CALL SetDrMd&(WINDOW(8),2)
 CALL RectFill&(WINDOW(8),231,21,394,31)
 WHILE testpat%<=8
   COLOR 1,0
   CALL SetDrMd&(WINDOW(8),1)
   Prin 399,50,"Druk op de linker muistoets"
   Prin 399,58,"voor (volgende) pattern"

   'leeg muisbuffer
   WHILE MOUSE(0)<>0:WEND
   WHILE MOUSE(0)=0:WEND
   LINE (399,42)-(630,70),0,bf

   COLOR 2:CALL RectFill&(WINDOW(8),21,21,189,149)
   COLOR 1,2:FOR i%=0 TO 11:Prin
25,30+i%*10,vergelijk$(i%):NEXT
   CALL SetDrMd(WINDOW(8),2)
   IF testpat%>0 THEN
     CALL
RectFill&(WINDOW(8),231,21+(testpat%-1)*10,394,31+(testpat%-1)
*10)
     CALL
RectFill&(WINDOW(8),231,21+testpat%*10,394,31+testpat%*10)
   END IF
   FOR i%=0 TO 11
     Match testpat$(testpat%),vergelijk$(i%)
     IF Matchfound% THEN
       CALL RectFill&(WINDOW(8),23,22+i%*10,187,32+i%*10)
     END IF
   NEXT
   testpat%=testpat%+1
  WEND

   Prin 420,70,"Einde demo.."
   CALL Delay&(200)
   CALL SetDrMd&(WINDOW(8),1)
   COLOR 1,0:CLS

RETURN
```

```
' BehandelMuis : Wanneer de sub FileRequester aktief is, zal
hier naar toe
' gesprongen worden wanneer er op de linkse muisbutton
gedrukt wordt.
' De routine zorgt voor de afhandeling van de verschillende
gadgets / klik-
' gebieden. Eveneens zorgt deze routine er voor dat er
vlaggen gezet wordt
' worden (wanneer dat nodig is) voor de onderbreking van "de
directory haal
' routine". Laatste verandering : 12 Oktober 1989
'============
BehandelMuis:

  'x-y coordinaten halen
  minfo%=MOUSE(0)
  mx%=MOUSE(3):my%=MOUSE(4)
  'Is er geklikt in het FileRequester window ?
  IF WINDOW(0)=2 THEN
    'Is er geklikt in de FileBox ?
    IF (mx%>=4 AND mx%<=246) AND (my%>=4 AND my%<=83) THEN
      GOSUB FileBoxKlik
    ELSE
      'Er is WAARSCHIJNLIJK geklikt op een
FileRequestergadget > Ga zoeken
      'welke dit is en zet dit in Klik% (0 indien niet op
een gadget geklikt)

      aktie%=0:gStart%=0:gEind%=GADGETS%-1
      GOSUB BehandelGadgets
      IF Klik%<>0 AND Getting% THEN StopRequest%=-1
    END IF
  END IF

RETURN


'FileBoxKlik kijkt of de klik in het FileBOX legaal is en zo
ja wordt de
'aangeklikte rij even opgelicht. Wanneer op een directory
geklikt wordt zal
'de LeesDir stopvlag gezet anders rijinhoud naar filenaam
kopiEren.
'===========
FileBoxKlik:

 'Bereken rij in de FileBox
 FileSelect%=INT((my%-4)/8)

 'Zijn er files in de FileBox EN staat er op de geklikte
plaats wel "iets" :
 '(Stel dat er maar 1 file op de disk staat > rest filebox
blijft dan leeg)
 IF top%+FileSelect%<Entry% AND Entry%<>0 THEN
   aktie%=0
   'Licht file even op
   CALL SetDrMd(WINDOW(8),2)
   CALL
RectFill(WINDOW(8),7,FileSelect%*8+4,242,FileSelect%*8+11)
   FOR i%=0 TO 800:NEXT
   CALL
RectFill(WINDOW(8),7,FileSelect%*8+4,242,FileSelect%*8+11)

   'Waneer dit een directory is (Kleur%(x%)=3) voor de
"directory haal routine"
   'een vlag zetten dat deze moet stoppen. Waneer het een
file is, deze gelijk
   'kopieren naar de box waar de aktieve Filenaam in staat.

   CALL SetDrMd(WINDOW(8),1)
   IF Kleur%(top%+FileSelect%)=3 THEN
     DirKlik%=-1:StopRequest%=-1
     in$=MID$(file$(top%+FileSelect%),7)
   ELSE
     LINE (39,118)-(264,125),0,bf
     e&=Move&(WINDOW(8),42,124)
     COLOR 1,0
     filename$=LEFT$(file$(top%+FileSelect%),27)
     CALL Text&(WINDOW(8),SADD(filename$),LEN(filename$))
     COLOR 1,2
   END IF
 END IF

RETURN

'BehandelGadgets zoekt aan de hand van de klikcoordinaten
MX% en MY% op
'welk gadget geklikt is. Hij zoekt in het bereik gstart% t/m
geind% welke
'het begin en eindnummer van het gadget aangeven.
'===============
BehandelGadgets:

 Klik%=0
 FOR i%=gStart% TO gEind%
  IF mx%=>gX%(i%) AND mx%<=gX1%(i%) THEN
    IF my%=>gY%(i%) AND my%<=gY1%(i%) THEN
Klik%=i%+1:i%=gEind%+1
  END IF
 NEXT

RETURN

'FileREQUESTER DATA
'========
FReqData:
```

```
DATA "DF0:","DF1:","DH0:","RAM:","VD0:","*[]?","/"
DATA 5,90,41,100,45,90,81,100,85,90,121,100,125,90,161,100
DATA
165,90,201,100,205,90,241,100,245,90,257,100,38,105,265,114
DATA
38,117,265,126,245,2,268,42,245,44,268,84,50,130,126,138,140,1
30,224,138

'PatternMatching DEMO data
'========================
PMdemoData:

'vergelijkingsdata
DATA "CInfo 1","CInfo 2", "Cinfo 3","Amiga 2000A","Amiga
2000B"
DATA "Daaag","Daag","Dag","Da","D","Model 1081","Model 1084S"

'patterndata
DATA "*[1-4]","*Info*","CInfo??","Amiga
2000[ab]","???","*a[a-z]*","*a?"
DATA "*i*[f1-9]*[2rxfoa-d]","CIn*[23]*"

SUB Prin(x%,y%,tekst$) STATIC
 e&=Move&(WINDOW(8),x%,y%):CALL
Text&(WINDOW(8),SADD(tekst$),LEN(tekst$))
END SUB

'FileRequest v1.0
'================
'Gekozen filenaam komt in :file$ te staan

SUB FileRequest(Wx%,Wy%,tekst$,scr%) STATIC
SHARED
Kleur%(),Device$(),file$(),Klik%,aktie%,CANCEL%,top%,filename$
,dir$,Entry%,FileSelect%,in$,StopRequest%,DirKlik%,file$,Match
found%,verg$,MAXFILES%,Getting%
  ON MOUSE GOSUB BehandelMuis
  lck&=0
  FileInfoBlock&=AllocMem&(260&,0)
  IF FileInfoBlock&=0 THEN ERROR 7 'GEHEUGEN PROBLEEM

  MOUSE OFF
  GOSUB WindowOpbouw
  aktie%=-1:EndRout%=-1:Klik%=0
  Chosen%=-1:StopRequest%=0

  MOUSE ON
  'De hoofdroutine "bestuurt de gadgets" en neemt aan de
hand van bepaalde
  'vlaggen handelingen. Wanneer er bv. vanuit subroutines de
vlag Aktie%
  'wordt gezet dan moet er een bepaalde handeling worden
verricht. Is EndRout%
  'gezet dan wordt uit de hoofdroutine gesprongen

'HoofdRoutine

  WHILE EndRout%

    WHILE aktie%
      SLEEP
    WEND

GadStopREQ:
    aktie%=-1
    'Op DF0:/DF1: etc. geklikt ?
    IF Klik% >=1 AND Klik% <=5 THEN
      dir$=Device$(Klik%-1)
      COLOR 1,0:LINE (39,106)-(264,113),0,bf
      Prin 42,112,dir$
      GOSUB LeesDir

    ELSEIF Klik%=6 THEN
      'MATCH gadget
      WINDOW 3,,(Wx%+20,Wy%+40)-(Wx%+200,Wy%+50),0,scr%
      Prin 5,12,"Match:"
      Invoer 50,56,6,116,1,2,0,1,verg$
      WINDOW CLOSE 3
      verg$=in$

    ELSEIF Klik%=7 THEN
      'Vorige directory halen (stuk van dir$ afhalen)
      'van rechts naar links: wordt een : gevonden dan
for-next afbreken
      'of wordt er een / gevonden en deze / is niet het
laatste teken dan
      'ook for-next afbreken. Bij afbreken wordt dan de
afbreekpositie
      'onthouden die dan het nieuwe einde van dir$ wordt.

      FOR i%=LEN(dir$) TO 1 STEP -1
        c%=ASC(MID$(dir$,i%,1))
        IF c%=58 OR (c%=47 AND i%<>LEN(dir$)) THEN
endstr%=i%:i%=0
      NEXT

      IF endstr%<>0 THEN dir$=LEFT$(dir$,endstr%) ELSE
dir$=":"
      COLOR 1,0:LINE (39,106)-(264,113),0,bf
      Prin 42,112,LEFT$(dir$,27)
      GOSUB LeesDir

    ELSEIF Klik%=11 OR Klik%=10 THEN
      '\/ gadget : laatste 10 in filebox (of minder dan 11
files) dan niks
      'doen, anders vorige 10 laten zien.
      herprint%=-1
      IF Klik%=11 THEN
        IF top%>=Entry%-10 THEN
          herprint%=0
        ELSE
          'aantal entrys nog te gaan ?
          lns%=Entry%-top%-10
          IF lns%>10 THEN lns%=10
          top%=top%+lns%
        END IF
      ELSEIF Klik%=10 THEN
        IF top%=0 THEN
          herprint%=0
        ELSE
          'aantal entrys nog te gaan ?
          lns%=top%
          IF lns%>10 THEN lns%=10
          top%=top%-lns%
        END IF
      END IF

      'Moet de inhoud van de filebox verandert worden ?
      IF herprint% THEN
        COLOR 2:CALL
RectFill&(WINDOW(8),5&,3&,245&,83&):COLOR 1,2
        FOR i%=top% TO top%+9
          COLOR Kleur%(i%)
          Prin 7,(i%-top%)*8+10,file$(i%)
        NEXT
      END IF

    ELSEIF Klik%=13 OR Klik%=12 THEN
      'Op OK of CANCEL gadget gedrukt .. en "uit
hoofdroutine" vlag zetten
      IF Klik%=13 THEN CANCEL%=-1 ELSE CANCEL%=0
      EndRout%=0

    ELSEIF Klik%=8 THEN
      'Nieuwe directorynaam intikken
      Invoer 100,38,106,216,1,0,0,1,dir$
      dir$=in$
      MOUSE ON
      GOSUB LeesDir

    ELSEIF Klik%=9 THEN
      'Nieuwe filenaam intikken
      MOUSE OFF
      Invoer 30,38,118,216,1,0,0,1,filename$
      filename$=in$
      MOUSE ON

    END IF

'De StopRequest / DirKlik vlag afhandelen
'Wanneer de StopRequest% vlag true is betekent dat: een klik
op een gadget of
'een klik op een directory in de filebox TERWIJL de
directory ingelezen werd.
'=========
VlagContr:
    IF StopRequest% THEN
      MOUSE OFF
      IF DirKlik% THEN
        DirKlik%=0:StopRequest%=0
        'directorynaam toevoegen aan bestaande directorynaam
in DirGadget
        IF RIGHT$(dir$,1)="/" OR RIGHT$(dir$,1)=":" THEN
          dir$=dir$+in$
        ELSE
          dir$=dir$+"/"+in$
        END IF
        'de nieuwe directorynaam in DirGadget zetten
        COLOR 1,0:LINE (39,106)-(264,113),0,bf
        Prin 42,112,LEFT$(dir$,27)

        MOUSE ON
        GOSUB LeesDir
        'nu kan er weer de StopRequest vlag gezet zijn dus
        'gelijk terugspringen naar VlagContr en controleren
        'om bepaalde redenen wordt hier een goto gebruikt
        GOTO VlagContr

      ELSE
        'er is tijdens het inlezen van de directory gedrukt
op een gadget
        StopRequest%=0
        MOUSE ON
        'nee, geen overbodige goto
        GOTO GadStopREQ
      END IF
    END IF

  Klik%=0
 WEND ' { HoofdRoutine }

 CALL FreeMem&(FileInfoBlock&,260&)
 CALL UnLock&(lck&)
 WINDOW CLOSE 2

 'de filename (dir$+filename$) aan elkaar koppelen. Het is
mogelijk heel de
 'pathstring (bv df0:s/startup-sequence) in het filegadget
te zetten (file-
 'name$) maar dan moet dirgadget wel leeg zijn (dir$="").
```

```
  IF NOT CANCEL% THEN
    IF RIGHT$(dir$,1)=":" OR RIGHT$(dir$,1)="/" THEN
      file$=dir$+filename$
    ELSE
      IF dir$="" THEN file$=filename$ ELSE
file$=dir$+"/"+filename$
    END IF
  END IF

 EXIT SUB

'============
WindowOpbouw:
  WINDOW
2,"Verzoek:"+tekst$,(Wx%,Wy%)-(Wx%+280,Wy%+140),22,scr%
  COLOR 2
  CALL RectFill&(WINDOW(8),5&,3&,245&,83&)
  CALL RectFill&(WINDOW(8),2&,133&,276&,138&)
  COLOR 3:CALL RectFill&(WINDOW(8),246,3,267,83)
  COLOR 1,2:LINE (4,2)-(246,84),,b
  LINE (245,2)-(268,84),,b
  LINE (245,43)-(268,43)
  AREA (252,41):AREA STEP(0,-23):AREA STEP(-5,0):AREA
STEP(10,-15)
  AREA STEP(10,15):AREA STEP(-5,0):AREA STEP(0,23):AREA
STEP(-10,0)
  AREAFILL
  AREA (252,45):AREA STEP(0,23):AREA STEP(-5,0):AREA
STEP(10,15)
  AREA STEP(10,-15):AREA STEP(-5,0):AREA STEP(0,-23):AREA
STEP(-10,0)
  AREAFILL
  top%=0
  IF Entry%=0 THEN
    Prin 25,30,"Kies het device waar u de"
    Prin 41,45,"inhoud van wilt zien."
  ELSE
    'files die nog in array zitten printen
    IF Entry%>9 THEN lns%=9 ELSE lns%=Entry%-1
    FOR i%=0 TO lns%
      COLOR Kleur%(i%)
      Prin 7,i%*8+10,file$(i%)
    NEXT i%
  END IF
  FOR i%=0 TO 6:GadghDraw (5+i%*40),90,Device$(i%):NEXT
  COLOR 1,0:Prin 2,112,"Dir":Prin 2,124,"File"
  LINE (38,105)-(265,114),,b
  LINE (38,117)-(265,126),,b
  GadghDraw 50,130," In orde ":GadghDraw 140,130," Afbreken "
  COLOR 1,0
  Prin 42,112,LEFT$(dir$,27)
  Prin 42,124,LEFT$(filename$,27)
RETURN

'=======
LeesDir:

 MOUSE OFF
 COLOR 2:CALL RectFill&(WINDOW(8),5&,3&,245&,83&)
 COLOR 1,2

 'Controleer op eventuele ongeldigheden
 IF lck& >0 THEN CALL UnLock&(lck&)
 lck&=Lock&(SADD(dir$+CHR$(0)),-2)
 IF lck&=0 THEN
   BEEP
   Prin 46,30,"    LOCK fout":Prin 46,40,"Directory/Device"
   Prin 46,50,"  bestaat niet"
 ELSE
   re&=Examine&(lck&,FileInfoBlock&)
   IF re&=0 THEN
     BEEP
     Prin 46,30,"Examine Fout":Prin 46,43," Vreemd ?!.."
   ELSE
     type&=PEEKL(FileInfoBlock&+4)
     IF type& <=0 THEN
       BEEP
       Prin 20,30,"DAAR trappen we niet in!"
       Prin 20,43,"De directory is een FILE!"
     END IF
   END IF
 END IF

 Entry%=0:top%=0:Linecnt%=0:Getting%=-1
 IF lck&=0 OR re&=0 OR type&<=0 THEN Getting%=0:MOUSE
ON:RETURN
 COLOR 1:CALL RectFill&(WINDOW(8),263,90,274,100)

 MOUSE ON
 'nu kunnen we weer onderbroken worden: onderbroken worden
tussen Entry%=0 en
 'bv top%=0 kan rare dingen tot gevolg hebben

 COLOR ,2
 'lees directory : naam uit FileInfoBlock halen, is dit een
directory dan
 'in kleurarray andere kleur zetten en de tekst "(DIR)"
ervoor zetten.
 'na filenaam ingelezen te hebben, deze laten zien, zonodig
scrollen.

 WHILE ExNext&(lck&,FileInfoBlock&) <> 0 AND Entry% <=
MAXFILES%  AND NOT StopRequest%
   EN$="":i%=1:Matchfound%=-1
   ascii%=PEEK(FileInfoBlock&+8)
     WHILE
 ascii%<>0:EN$=EN$+CHR$(ascii%):ascii%=PEEK(FileInfoBlock&+8+i%
):i%=i%+1:WEND
     IF verg$<>"" AND PEEKL(FileInfoBlock&+4) <= 0 THEN
       Match verg$,EN$
     END IF
     IF Matchfound% THEN
       IF PEEKL(FileInfoBlock&+4) > 0 THEN
Kleur%(Entry%)=3:file$(Entry%)="(DIR) "+EN$ ELSE
Kleur%(Entry%)=1:file$(Entry%)=EN$
       MOUSE OFF
       COLOR Kleur%(Entry%)
       IF Linecnt%=10 THEN
         CALL ScrollRaster&(WINDOW(8),0,8,5,4,245,83)
         Prin 7,82,file$(Entry%):top%=top%+1
       ELSE
         Prin 7,Linecnt%*8+10,file$(Entry%)
         Linecnt%=Linecnt%+1
       END IF
       Entry%=Entry%+1
       MOUSE ON
     END IF
   WEND
   COLOR 0:CALL RectFill&(WINDOW(8),263,90,274,100)
   Getting%=0

 RETURN

 END SUB

 SUB GadghDraw (x%,y%,T$) STATIC
   plus%=LEN(T$)*8:COLOR 3:CALL
RectFill&(WINDOW(8),x%+1,y%+1,x%+plus%+3,y%+9)
   LINE (x%,y%)-(x%+plus%+4,y%+10),1,b:COLOR
1,3:e&=Move&(WINDOW(8),x%+3,y%+8)
   CALL Text&(WINDOW(8),SADD(T$),LEN(T$))
 END SUB

'Invoer V2.0
'===============
'Ingevoerde tekst komt in : in$ te staan
'Is er een getal ingevoerd dan komt het getal in in! te staan
' *Invoertype 0:alles toegestaan,   1: 0123456789-.*^/()+=
'             2:cijfers en letters, 3: jnJNyY,   4:
0123456789.
' *Ret(urn)    0:bij leeg veld druk op return niet toegestaan
'              1:druk op return altijd toegestaan
'Speciale toetsen: F9 = vorig woord,       F10 = Volgend
woord
'            Cursor /\ = begin tekst, Cursor \/ = eind tekst
'                   ESC = Wis alle tekst
'Scrollbare edit routine

SUB
Invoer(limiet%,xpos%,ypos%,breedte%,Voorgrond%,Achtergrond%,ty
pe%,ret%,buf$) STATIC
 SHARED in$,in,csrbeeld%

 GOSUB VoorBereiding

Toets:
 key$=""
 WHILE key$="":key$=INKEY$:WEND
 key%=ASC(key$)

 IF key%=13 THEN
   'Controleer of drukken op return toegestaan is als veld
leeg is,
   'is het veld niet leeg dan uit de sub
   IF strp%=0 AND ret%=0 THEN
     BEEP
   ELSE
     PUT (xpo%,ypo%),CursorBeeld%,XOR:in=VAL(in$)
     EXIT SUB
   END IF

 ELSEIF key%=8 THEN
   'Alleen deleten wanneer veld niet leeg is. Als dit zo is
alleen resultaat
   'van delete aktie laten zien wanneer cursor niet geheel
links staat en
   'dan moet er op worden gelet of de cursor niet achter het
laatste teken
   'staat (insert%=0) en of het laatste teken niet op de
meeste rechtse
   'plaats stond in de box OMDAT er dan een spatie
toegevoegd moet worden
   'aan de printstring.
   IF strp%>0 THEN
     buf$=LEFT$(in$,strp%-1)
     rest$=MID$(in$,strp%+1)
     in$=buf$+rest$
     strp%=strp%-1
     IF xpo%>MinX% THEN
       csr%=csr%-1
       prn$=LEFT$(rest$,MaxBOXchar%-csr%)
       PUT (xpo%,ypo%),CursorBeeld%,XOR
       IF insert%=0 OR strp%+MaxBOXchar%-csr%>LEN(in$) THEN
prn$=prn$+" "
       xpo%=xpo%-8
       Prin xpo%,ypos%,prn$
       PUT (xpo%,ypo%),CursorBeeld%,XOR
     END IF
   END IF

 ELSEIF key%=127 AND strp%<>limiet% THEN
```

```
   'DEL toets: teken er tussen uit halen en oppassen bij
printen: als
   'meest rechtse teken het laatste teken was dan spatie
toevoegen
   '(aan printstring)

   PUT (xpo%,ypo%),CursorBeeld%,XOR
   in$=LEFT$(in$,strp%)+MID$(in$,strp%+2)

   'je hoeft niet te printen wanneer cursor helemaal rechts
staat
   IF csr%<>MaxBOXchar% THEN
     prn$=MID$(in$,strp%+1,MaxBOXchar%-csr%)
     IF csr%+LEN(prn$)<MaxBOXchar% THEN prn$=prn$+" "
     Prin xpo%,ypos%,prn$
   END IF
   PUT (xpo%,ypo%),CursorBeeld%,XOR

 ELSEIF key%=31 THEN
   IF strp%>0 THEN
     'Cursor links: Als cursor op meest linkse positie stond
dan scrollen
     strp%=strp%-1
     insert%=1
     IF xpo%=MinX% THEN
       PUT (xpo%,ypo%),CursorBeeld%,XOR
       Prin xpo%,ypos%,MID$(in$,strp%+1,MaxBOXchar%)
       PUT (xpo%,ypo%),CursorBeeld%,XOR
     ELSE
       srt%=-8:GOSUB CsrAanUit
       xpo%=xpo%-8:csr%=csr%-1
     END IF
   END IF

 ELSEIF key%=30 THEN
   'Cursor rechts : wanneer de cursor voor het laatste teken
staat dan aktie
   'op cursor rechts welke is: als de cursor wordt
verplaatst naar het laatste
   'teken dan "insert" mode uitzetten. STOND de cursor
geheel rechts dan
   'moet er gescrolld worden, anders cursor enkel
verplaatsten.
   IF strp%<LEN(in$) THEN
     strp%=strp%+1
     IF strp%=LEN(in$) THEN insert%=0
     IF xpo%=MaxX% THEN
       PUT (xpo%,ypo%),CursorBeeld%,XOR
       Prin
MinX%,ypos%,MID$(in$,1+strp%-MaxBOXchar%,MaxBOXchar%)
       PUT (xpo%,ypo%),CursorBeeld%,XOR
     ELSE
       srt%=8:GOSUB CsrAanUit
       xpo%=xpo%+8:csr%=csr%+1
     END IF
   END IF

 ELSEIF key%=27 THEN
   'ESCAPE : wis de ingevoerde tekst
   PUT (xpo%,ypo%),CursorBeeld%,XOR
   csr%=0:strp%=0:in$="":LINE
(MinX%,ypo%)-(MaxX%,ypo%+7),Achtergrond%,bf
   xpo%=MinX%:PUT (xpo%,ypo%),CursorBeeld%,XOR

 ELSEIF key%=28 THEN
   'Cursor begin
   PUT (xpo%,ypo%),CursorBeeld%,XOR
   csr%=0:strp%=0
   IF LEN(in$)<MaxBOXchar% THEN
spatie$=SPACE$(MaxBOXchar%-LEN(in$)) ELSE spatie$=""
   xpo%=MinX%:Prin MinX%,ypos%,LEFT$(in$,MaxBOXchar%)+spatie$
   PUT (xpo%,ypo%),CursorBeeld%,XOR

 ELSEIF key%=29 THEN
   'Cursor einde
   PUT (xpo%,ypo%),CursorBeeld%,XOR
   csr%=LEN(in$):strp%=csr%:insert%=1
   IF csr%>MaxBOXchar% THEN csr%=MaxBOXchar%
   xpo%=MinX%+csr%*8
   IF LEN(in$)<MaxBOXchar% THEN
spatie$=SPACE$(MaxBOXchar%-LEN(in$)) ELSE spatie$=""
   Prin MinX%,ypos%,RIGHT$(in$,MaxBOXchar%)+spatie$
   PUT (xpo%,ypo%),CursorBeeld%,XOR

 ELSEIF key%=137 THEN
   'zoek vorig woord : eerst alle spaties links van cursor
overslaan
   'daarna begin woord gaan zoeken (spatie zoeken)
   zps%=strp%:csrm%=0:zoekdoor%=-1
   WHILE zps%=>1 AND zoekdoor%
     IF MID$(in$,zps%,1)=" " THEN
       zps%=zps%-1:csrm%=csrm%-1
     ELSE
       zoekdoor%=0
     END IF
   WEND
   IF zps%=>1 THEN
     zoekdoor%=-1
     WHILE zps% AND zoekdoor%
       IF MID$(in$,zps%,1)<>" " THEN
         zps%=zps%-1:csrm%=csrm%-1
       ELSE
         zoekdoor%=0
       END IF
     WEND
     ss%=zps%
     GOSUB PrnWrd
   END IF

 ELSEIF key%=138 THEN
   'zoek volgend woord : zoek spatie en sla de spaties na
die spatie
   'over om zo bij het begin van het woord te komen.

   zps%=strp%+1:csrm%=1
   WHILE MID$(in$,zps%,1)<>" " AND zps%<=LEN(in$)
     zps%=zps%+1:csrm%=csrm%+1
   WEND
   IF zps%<=LEN(in$) THEN
     WHILE MID$(in$,zps%+1,1)=" " AND zps%+1<=LEN(in$)
       zps%=zps%+1:csrm%=csrm%+1
     WEND
     ss%=zps%
     GOSUB PrnWrd
   END IF

 ELSE
   'controleer of teken toegestaan is
   IF st%=1 THEN
     IF (key%<Lo% OR po%>hi%) AND (key%<11% OR key%>12%)
THEN Toets
   ELSE
     IF INSTR(st$,key$)=0 THEN Toets
   END IF

   IF strp%=limiet% THEN Toets
   strp%=strp%+1
   IF csr%<MaxBOXchar% THEN csr%=csr%+1
   PUT (xpo%,ypo%),CursorBeeld%,XOR

   IF insert%=0 THEN
     '(cursor achter laatste teken) Is de cursor uiterst
rechts en zijn er
     '"genoeg tekens" om te scrollen dan scrollen, anders 1
teken printen
     in$=in$+key$
     IF strp%>MaxBOXchar% AND xpo%=MaxX% THEN
       Prin
MinX%,ypos%,MID$(in$,1+strp%-MaxBOXchar%,MaxBOXchar%)
     ELSE
       e&=Move&(WINDOW(8),xpo%,ypos%)
       CALL Text&(WINDOW(8),SADD(key$),1)
       xpo%=xpo%+8
     END IF

   ELSE
     'kijken of we met extra teken niet over de lengtelimiet
heen gaan en
     'zo nee dan teken invoegen.
     IF LEN(in$)+1>limiet% THEN
       strp%=strp%-1
       IF csr%-1<MaxBOXchar% THEN csr%=csr%-1
     ELSE
       buf$=MID$(in$,strp%,LEN(in$)):in$=LEFT$(in$,strp%-1)
       in$=in$+key$+buf$
       Prin MinX%,ypos%,MID$(in$,1+strp%-csr%,MaxBOXchar%)
       IF xpo%<>MaxX% THEN xpo%=xpo%+8
     END IF
   END IF

   PUT (xpo%,ypo%),CursorBeeld%,XOR

 END IF

 GOTO Toets

'Voorbereiding: De al "ingevoerde tekst" printen,
cursorpositie goed zetten,
' kleuren instellen, Cursor aanmaken als deze nog niet
gemaakt is (in
' csrbeeld% array) en juiste toegestane invoerkarakters
kiezen.
'============
VoorBereiding:

 in$=buf$:strp%=LEN(buf$)

MaxBOXchar%=breedte%/8:MinX%=xpos%+4:MaxX%=MaxBOXchar%*8+MinX%

 COLOR Voorgrond%,Achtergrond%
 IF LEN(buf$)=>MaxBOXchar% THEN
```

```
    csr%=MaxBOXchar%
    Prin MinX%,ypos%+6,RIGHT$(in$,MaxBOXchar%)
  ELSE
    csr%=LEN(buf$)
    Prin MinX%,ypos%+6,in$+SPACE$(MaxBOXchar%-strp%)
  END IF

  xpos%=MinX%+8*csr%
  xpo%=xpos%:ypo%=ypos%
  ypos%=ypos%+6
  'vanaf hier ypos%=ypo%+6 (wordt veel gebruikt in Prin x,y,$)

  IF csrbeeld% THEN
    PUT (xpo%,ypo%),CursorBeeld%,XOR
  ELSE
    LINE (xpo%,ypo%)-STEP(1,7),Voorgrond% XOR Achtergrond%,b
    GET (xpo%,ypo%)-(xpo%+1,ypo%+7),CursorBeeld%
  END IF
  csrbeeld%=-1

 IF type%=0 THEN
    st%=1:Lo%=32:hi%=255:l1%=Lo:l2%=hi%
  ELSEIF type%=1 THEN
    st%=0:st$="0123456789-.*/^()+="
  ELSEIF type%=2 THEN
    st%=1:Lo%=48:hi%=57:l1%=65:l2%=122
  ELSEIF type%=3 THEN
    st%=0:st$="jnJNyY"
  ELSEIF type%=4 THEN
    st%=0:st$="0123456789."
 END IF

RETURN ' { Voorbereiding }

CsrAanUit:
 PUT (xpo%,ypo%),CursorBeeld%,XOR
 PUT (xpo%+srt%,ypo%),CursorBeeld%,XOR
RETURN

PrnWrd:
 IF ss%=LEN(in$) THEN insert%=0 ELSE insert%=1
 'kijken of de inhoud van de box opnieuw geprint moet worden
 PUT (xpo%,ypo%),CursorBeeld%,XOR
 IF csr%+csrm%=>MaxBOXchar% OR csr%+csrm%<0 THEN
   csr%=0:strp%=ss%

prn$=MID$(in$,1+strp%,MaxBOXchar%):prn$=prn$+SPACE$(MaxBOXchar
%-LEN(prn$))
   Prin MinX%,ypos%,prn$
   xpo%=MinX%
 ELSE
   csr%=csr%+csrm%:xpo%=MinX%+csr%*8:strp%=ss%
 END IF
 PUT (xpo%,ypo%),CursorBeeld%,XOR

RETURN

END SUB ' { Invoer }

SUB Pijl (x%,y%,height%,b%) STATIC
 AREA (x%,y%):AREA STEP(b%,height%/2):AREA
STEP(-b%,height%/2):AREA STEP (0,-height%):AREAFILL
END SUB

SUB title (x%,y%,title$,wwidth%) STATIC
 COLOR 3:CALL RectFill&(WINDOW(8),0,0,wwidth%,8)
 COLOR 1,3:LINE (0,9)-(wwidth%,9)
 e&=Move&(WINDOW(8),x%,y%)
 CALL Text&(WINDOW(8),SADD(title$),LEN(title$))
END SUB


  ' STRING PATTERN MATCHING Module
  ' *Datum: 20-Aug-1989  *Versie v1.0
  ' Geschreven door R.Sloot
  ' gebruikte pattern karakters: *,[] and ?
  ' *end,*end[1-4] = *end[1234],*end[16-9] = *end[16789],
*bla[m-z]?bla*
  ' start*end???,?????,???lala*??boe

'--------------------------------------------------------------
--------
  ' De routine keert met MatchFound% = TRUE (-1) terug als er
een over-
  ' eenkomst en bij geen overeenkomst MatchFound% = FALSE (0)
  ' Opmerking: Match is niet case-sensitive, maar is hij wel
te maken
  ' door de ucase$ statements (2) uit de routine te halen.
  ' Aanroep: match "st[a-b]rt*[0-9].INFO",vergelijking$

SUB Match (pat$,comp$) STATIC
 SHARED Matchfound%
 'watch% :vlag voor uitzonderingsgeval pattern karakters
 'ppos % :positie in de pattern$
 'cpos % :positie in vergelijkingsstring waar tot daar aan
toe overeenkomst
 '    p % :positie van (volgend) patternkarakter

 watch%=0:pat$=UCASE$(pat$):comp$=UCASE$(comp$)
 Matchfound%=-1:ppos%=1:GOSUB NextPATChar:ppos%=p%

 IF ppos%=>1 THEN
    'vergelijk tekst als eerste deel tekst is
    IF ppos%=>2 THEN
       IF LEFT$(pat$,ppos%-1)<>LEFT$(comp$,ppos%-1) THEN
Matchfound%=0:EXIT SUB
    END IF
    cpos%=ppos%

    'vergelijk totdat pat$ doorlopen is
    WHILE ppos%<=LEN(pat$)
      pchar$=MID$(pat$,ppos%,1)
      IF pchar$="*" THEN
         'is * het laatste teken dan overeenkomst
         IF ppos%=LEN(pat$) THEN EXIT SUB
         ppos%=ppos%+1:GOSUB
NextPATChar:p1%=INSTR(ppos%,pat$,"*")

         'als er een ?,* of [ direkt achter * staat is dat
een uitzonderings-
         'geval. Staat er een * niet direkt achter * dan
tussenliggende tekst
         'vergelijken, anders (er staat een [ of ? niet
direkt achter *)
         'moet het laatste stuk van de vergelijkingstring
(comp$) vergeleken worden

         IF ppos%=p% THEN
            watch%=-1
         ELSEIF p1%>0 THEN
            search$=MID$(pat$,ppos%,p%-ppos%)
            p1%=INSTR(cpos%,comp$,search$)
            IF p1%=0 THEN Matchfound%=0:EXIT SUB ELSE
cpos%=p1%+LEN(search$)
         ELSE
            IF p%=0 THEN p%=LEN(pat$)+1
            search$=MID$(pat$,ppos%,p%-ppos%)

            'nu cpos% op goede positie van rechts gezien
zetten, waarvoor het
            'aantal werkelijke karakters vanaf ppos% nog
geteld moeten worden.
            'bv :*test[1-4]? :aantal werkelijke karakters na
* = 6 ([--] en ?
            'tellen als 1) dus cpos% moet op 6e karakter van
rechts gezet worden

            GOSUB RightCsearch:IF Matchfound%=0 THEN EXIT SUB
            IF search$<>MID$(comp$,cpos%,LEN(search$)) THEN
Matchfound%=0:EXIT SUB
            cpos%=cpos%+LEN(search$)
         END IF
         ppos%=p%
      ELSEIF pchar$="[" THEN
         '[---] ontleden
         ppos%=ppos%+1
         p%=INSTR(ppos%,pat$,"]")
         p1%=INSTR(ppos%,pat$,"-")
         IF p1%=0 OR p1%=>p% THEN p1%=p%
         search$=MID$(pat$,ppos%,p1%-ppos%)
         endb$=MID$(pat$,p%-1,1)

         IF watch% THEN
            'als er geen * meer is (*[1-4]abc?) dan
uitzoeken welk teken
            'van rechts gezien in comp$ vergeleken moet
worden (RightCsearch)
            'met bereik anders (*[a-d]2500 UX*) teken zoeken
in comp$ vanaf
            'cpos% dat voldoet aan bereik.


watch%=0:in%=0:p1%=1:ppos%=p%+1:p%=INSTR(ppos%,pat$,"*")
            IF p%=0 THEN
               GOSUB RightCsearch:IF Matchfound%=0 THEN EXIT
SUB
               c$=MID$(comp$,cpos%,1):cpos%=cpos%+1
               IF NOT (INSTR(search$,c$)=>1 OR
(c$=>RIGHT$(search$,1) AND c$<=endb$)) THEN
Matchfound%=0:EXIT SUB
            ELSE
               notfound%=-1
               WHILE notfound% AND cpos%<=LEN(comp$)
                  c$=MID$(comp$,cpos%,1)
                  IF INSTR(search$,c$)=>1 OR
(c$=>RIGHT$(search$,1) AND c$<=endb$) THEN notfound%=0
                  cpos%=cpos%+1
               WEND
               IF notfound% THEN Matchfound%=0:EXIT SUB
            END IF
         ELSE
            '1 teken op cpos% in comp$ op bereik testen bv
a[ga-d]zo*
            IF cpos%>LEN(comp$) THEN Matchfound%=0:EXIT SUB
            c$=MID$(comp$,cpos%,1)
            IF NOT (INSTR(search$,c$)=>1 OR
(c$=>RIGHT$(search$,1) AND c$<=endb$)) THEN
Matchfound%=0:EXIT SUB
            ppos%=p%+1:cpos%=cpos%+1
            IF p%=LEN(pat$) AND cpos%<=LEN(comp$) THEN
Matchfound%=0:EXIT SUB
         END IF
      ELSEIF pchar$="?" THEN
            IF watch% THEN
               'uitzondering : *???abc[o-r]
               watch%=0:p%=INSTR(ppos%,pat$,"*"):IF p%=0
THEN p1%=0
               GOSUB RightCsearch:IF Matchfound%=0 THEN EXIT
SUB
            END IF
            'loop pat$ af : als het een ? is dan kijken of
er nog een teken
```

```
             'beschikbaar is in comp$, wordt er een [ or *
gevonden dan wordt
             'deze afgehandeld in de "hoofdloop" maar losse
tekens worden wel
             'afgehandeld in deze for-next

             FOR i%=ppos% TO LEN(pat$)
                p%=i%:c%=ASC(MID$(pat$,i%,1))
                IF c%=42 OR c%=91 THEN
                   i%=LEN(pat$)+10
                ELSEIF c%=63 THEN
                   IF cpos%>LEN(comp$) OR (i%=LEN(pat$) AND
cpos%<>LEN(comp$)) THEN Matchfound%=0:EXIT SUB
                   cpos%=cpos%+1
                ELSE
                   IF NOT c%=ASC(MID$(comp$,cpos%,1)) THEN
Matchfound%=0:EXIT SUB
                   cpos%=cpos%+1
                END IF
             NEXT
             IF i%=LEN(pat$)+11 THEN ppos%=p% ELSE ppos%=p%+1
      ELSE
         'tekststring vergelijken
         GOSUB NextPATChar
         IF p%=0 THEN p1%=LEN(pat$)+1 ELSE p1%=p%
         search$=MID$(pat$,ppos%,p1%-ppos%)
         IF search$<>MID$(comp$,cpos%,LEN(search$)) THEN
Matchfound%=0:EXIT SUB
         ppos%=p1%:cpos%=cpos%+LEN(search$)
      END IF
    WEND
 ELSE
    IF pat$<>comp$ THEN Matchfound%=0
 END IF

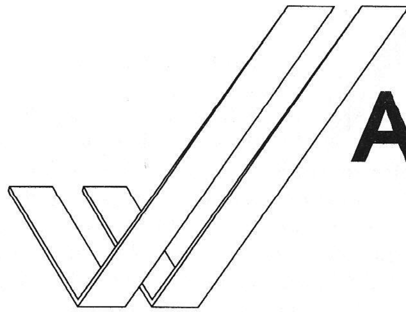
EXIT SUB 'de routines hieronder niet uitvoeren..
```

```
'NextPATChar : zoekt volgend pattern karakter : * [(]) of ?
'===========
NextPATChar:
 p%=0
 FOR i%=ppos% TO LEN(pat$)
    IF INSTR("*[?",MID$(pat$,i%,1))=>1 THEN
p%=i%:i%=LEN(pat$)+1
 NEXT
RETURN

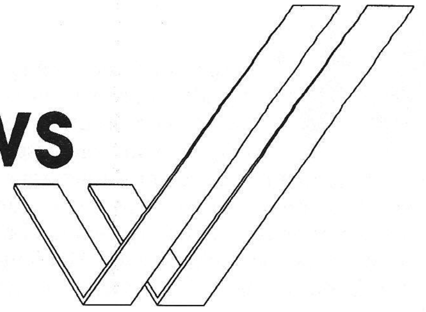
'RightCsearch : telt aantal werkelijke tekens vanaf ppos% in
pat$ en
'zet cpos% dan op goede positie van waaraf vergeleken moet
worden
'============
RightCsearch:
 in%=0
 FOR i%=ppos% TO LEN(pat$)
   c$=MID$(pat$,i%,1)
   IF c$="[" OR in% THEN
     IF c$="]" THEN p1%=p1%+1:in%=0 ELSE in%=-1
   ELSE
     p1%=p1%+1
   END IF
 NEXT
 cpos%=1+LEN(comp$)-p1%
 IF cpos%<1 THEN Matchfound%=0
RETURN

END SUB
```

Einde listing filereq



Amiga Nieuws in het kort



Bridgeboard voor A500

Eindelijk is het dan zover. Binnenkort kunnen Amiga 500-bezitters op normale snelheid genieten van MS-DOS software. Kolf Computer Supplies (KCS), bekend van de Power Cartridge voor de C-64, is nog druk bezig de laatste hand te leggen aan een Bridgeboard voor de Amiga 500.
Zij die nu een A500 hebben en MS-DOS willen draaien, zijn genoodzaakt om eerst de emulatie-software te draaien. Daarna wordt het (MS-) DOS ingevoerd en is de Amiga klaar voor gebruik. Maar de snelheid laat nogal te wensen over en de machine gaat vaak 'hangen'; een Guru is het gevolg. Dat houdt in dat de hele procedure weer van voren af aan begint. De lol is er op die manier al gauw af.
KCS is een aantal maanden geleden begonnen met de ontwikkeling van een soort Bridgeboard, dat we al kennen van de A2000. Zelf noemen ze het 'Power PC Board. De kaart is net zo groot als een interne geheugenuitbreiding en past exact onderin de Amiga 500. Het voordeel hiervan is dat die niet open geschroefd hoeft te worden; 'insteken en klaar!' is hier de leus.
De kaart zelf is voorzien van een V30-8 Mhz processor, 1 megabyte geheugen en een clock. De clock loopt op batterijen en is zowel voor de PC- als de Amiga-kant bruikbaar. Dat geldt in zekere zin ook voor het geheugen. Van de 1 MB is de helft geschikt voor de Amiga. Daarmee is de kaart wat dat betreft gelijk aan een interne geheugenkaart van 512 KB. De andere helft is ook bruikbaar, maar alleen als RAM-disk. Software hiervoor wordt bijgeleverd.
Alle aangesloten diskdrives zijn te gebruiken als MS-DOS drive. Die compatibiliteit geldt ook voor alle aansluitingen als parallelle- en seriële poort. Verder ondersteunt de kaart de beeldscherm modi MDA, Hercules en CGA.
Het voordeel van de kaart is het stukje software dat toch nog nodig blijft om van de Amiga een PC te maken. Door dat zo te laten, kan KCS het Power PC-Board steeds blijven uitbreiden. De kaart krijgt namelijk van de floppy instructies hoe zij zich moet gedragen. Zo staat de ondersteuning van de Commodore harddisk A590 ook al op het programma. Hetzelfde geldt voor EGA- en VGA-ondersteuning. In een latere versie zal de software daarvoor worden aangepast.
Het Power PC-Board wordt compleet geleverd met MS-DOS 4.01, GW-Basic, DOS Shell en drie Nederlandse boeken.
Inl: 078-310931

WordPerfect 4.1 vernieuwd

Op PC gebied verschijnt de ene versie na de andere van het overbekende tekstverwerkingspakket WordPerfect. En hoewel menig Amiga-gebruiker versie 5.0 al verwacht, vanwege de grafische mogelijkheden, blijft het maar bij 4.1. Desalniettemin werd deze versie behoorlijk vernieuwd. Je heb dan ook de neiging om te spreken van '4.2'.
Voor hen die al weer gewend waren aan de Nederlandse versie, betekenen de vernieuwingen een kleine teruggang, aangezien het programma weer in het Engels werkt, net als de eerste versie voor de Amiga.
Voor het eerst is het mogelijk om de buffer-grootte aan te geven, waardoor er op het geheugen kan worden bespaard. Verder kan men nu vanuit WP direct MS-DOS files lezen en schrijven; handig voor iemand die met beide systemen moet werken. Voorheen moest zoiets met een 'vreemde' utility gebeuren, zoals DOS-2-DOS.
Onder andere voor deze mogelijkheid is de file-functie (onder F5) helemaal veranderd en lijkt nu een beetje op de file-manager uit de WP Library. (Die file-manager doet op zijn beurt weer denken aan een programma als Disk-Wizard.) Bij exporteren van een file komt men weer in een soort bestandenlijst. Daarnaast kan er een overzicht worden opgevraagd van alle drives en directorie's en is sorteren op naam mogelijk.
Kleuren instellen hoeft nu niet meer via de Preferences. Met een klein mengpaneeltje binnen WordPerfect kunnen die naar wens worden ingesteld.
De paginalengte is nu ook te definiëren: er kan gekozen worden tussen 11 of 14 inch. De vervelende requester bij het afbreekstreepje (alleen maar weg te krijgen door aan te geven waar hij moest en 'OK') is vervangen door een comfortabel menuutje, waarin ook kan worden gekozen voor 'niet afbreken'.
De vernieuwde versie ondersteunt 283 printers en is gratis voor geregistreerde gebruikers van WordPerfect voor de Amiga.
Inl: 010-4070100

Portable Amiga

De firma Gigatron in Duitsland dacht het op tijd voor elkaar te kunnen hebben: een eigen ontwikkelde portable Amiga.
De bedoeling was dit unieke model op de grote beurs in Köln, de Amiga '89, te kunnen laten zien. Maar helaas voor de toeschouwers lukte dit niet op tijd.
Volgens Gigatron was het echt niet mogelijk het apparaat al in november klaar te hebben. Tijdens de Amiga '89 maakte het bedrijf bekend dat de laptop nu voor eind maart staat gepland.
Een woordvoerder van Gigatron vertelde ons desgevraagd dat zij momenteel geen enkele mededeling rond de laptop kunnen doen. Wel zij hij dat de machine naar aller waarschijnlijkheid op de Cebit in Hannover (21-28 maart) zal worden geïntroduceerd.

Amiga in Amersfoort

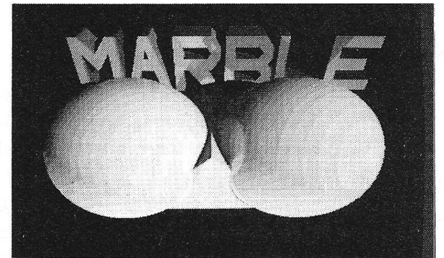
Werken met de Amiga mondt voor een groot aantal mensen uit in het volgen van cursussen, om op die manier nog meer uit het apparaat te kunnen halen. Inwoners van Amersfoort en omgeving zijn wat dat betreft in een bevoorrechte positie. Het Kultureel Centrum De Hof in Amersfoort (in de Flint) organiseert namelijk cursussen op het gebied van animatie en DTP (DeskTop Publishing).
Voorafgaand aan het 'echte' werk krijgt iedere cursist een introductie, waarbij wordt gewerkt aan de hand van DeLuxe Paint, de WorkBench en DigiView.
Bij het animeren staan de programma's Videoscape, PageFlipper en DeLuxe Paint ter beschikking en DTP'en gebeurt met DeLuxe Paint, PageSetter en Professional Page. Ook is het mogelijk alleen de introductie te volgen. Voor bedrijven bestaat daarnaast nog de mogelijkheid een cursus op maat te nemen.
Naast de cursussen kan iedere Amiga-bezitter tegen betaling gebruik maken van de faciliteiten van het Kultureel Centrum. Voor velen zal dit een uitkomst zijn, omdat apparatuur als laserprinter, dia-converter, etc. bijna nooit thuis voorhanden zijn.
Een prijsindicatie voor het overzetten van IFF-plaatjes naar dia's: *f* 30,- startkosten, en *f* 5,- tot *f* 7,50 per dia.
Inl: 033-729094

Van Spelcomputer tot high-end Black Tower

Je kunt het zo gek niet bedenken of het is wel mogelijk met een Commodore Amiga.' Dat zou een bezoeker bij 3Gitaal in Amsterdam kunnen denken. Dit bedrijf maakt met behulp van Amiga's en daarvoor geschikte apparatuur animaties voor onder andere de KLM, de Vara en RTL Veronique. Maar 3Gitaal is ook Commodore-dealer, en levert daarnaast nog andere interessante produkten voor de Amiga van andere leveranciers.
Wie wel eens meer wil dan een standaard Amiga en het geld er voor over heeft, kan eens overwegen of hij/zij de machine niet uit zal breiden met bijvoorbeeld een Bridgeboard, een processorversneller of een geheugenuitbreiding. Dat uitbreiden kan op verschillende manieren en bij meerdere dealers in Nederland en daarbuiten.
Een van die leveranciers is 3Gitaal in Amsterdam. Op de prijslijst van deze firma staat vrijwel net zo vaak 'Commodore' als 'X-Pert' als merk vermeld. En dat klopt ook, want 2Gitaal is importeur van dit Duitse merk.
X-Pert staat bekend als Amiga-tuner, wat zoiets wil zeggen als 'mooimaker'. Erg fraai om te zien is de Amiga Black Tower. Het een totaal omgebouwde Amiga 2000B met alle snufjes die je je maar kunt wensen. De ombouw bestaat uit een zwarte metalen kast met een zogenoemde servicedeur die gemakkelijk is voor het in- en uitbouwen van uitbreidingen van welke aard dan ook.
Het moederbord is voorzien van een IBM- geheugen met de nieuwe Fat Agnus chip. In tegenstelling tot de normale Amiga 2000 is er bij de Black Tower een display in de kast gebouwd die de snelheid in Megaherz aangeeft. Verder behoren een dubbele ventilator en een dubbele 230 volt-aansluiting voor het doorlussen van monitor en printer tot de standaard Black Tower-uitvoering.
In de kast is meer dan voldoende ruimte voor extra harddidsks en floppy-drives. Net als de kast zijn de muis en het toetsenbord in zwart uitgevoerd. 3Gitaal kan de Black Tower geheel naar de wensen van de klant samenstellen. De prijs van de standaarduitvoering *f* 5.998,-.
Een ander bijzonder item zijn de Cheaper Net- en de Apple Talk-netwerkkaarten. Daarmee kunnen Amiga's in een eigen net worden gekoppeld, maar ook aan een al bestaand netwerk. Zo kan met de Apple Talk-netwerkkaart de Amiga ook onder A-max (Apple Macintosh emulator voor de Amiga) gebruikt worden.
Maar ook voor de PC-kant van de Amiga heeft 3Gitaal diverse produkten van X-Pert naast die van Commodore. De X-Pert Bridgeboards zijn qua prestaties beter, maar daardoor wel duurder. Voor de bezitters van de Commodore Bridgeboards heeft 3Gitaal ook up-grade kits naar de types van X-Pert. Naast deze 'specialities' heeft 3Gitaal ook een upgrade service naar onder meer Kickstart 1.3, Fat Agnus, WorkBench 1.3 en Janus 2.0.
Tenslotte kan iedereen die met zijn/haar Amiga met video, plaatjes of iets dergelijks aan de slag wil bij 3Gitaal terecht voor genlocks, digitizers, scanners, image recorders en tekentablets. Natuurlijk hoort hier ook software bij. Op dit moment is 3Gitaal bezig met het aansluiten en gebruiksklaar maken van een CD-ROM aan de Amiga. Wie kwaliteit wil voor zijn/haar Amiga en graag naderhand van een stukje support wil genieten, moet eens een kijkje nemen bij 3Gitaal in Amsterdam. Vooral de professionele gebruiker vindt hier alles mogelijk voor een top-Amiga.
Inl: 020-970035

9 MB intern in A500

Niemand hield het voor mogelijk: een geheugenuitbreiding van 8 MB in een Amiga 500. Progressive Peripheral & Software (PPS) uit Denver (VS) is dat gelukt en het resultaat heet de EXP-8000+. Voorheen werden A500-gebruikers al verrast door de interne kaarten die tot 2 MB RAM gaven, bijvoorbeeld de Minimax van de Duitse firma Gigatron.



De EXP-8000+ kaart van de Amerikaanse firma PPS is ook bedoeld voor inbouw in de Amiga. Alleen de montage gaat iets anders in zijn werk dan bij de normale interne kaarten. De kaart wordt er dan van de bovenkant ingemonteerd, zodat de klep aan de onderkant gewoon gesloten blijft. Daardoor blijft de mogelijkheid bestaan nog een normale 512 KB kaart te monteren. In totaal komt het geheugen daarmee op 9 MB intern. Het enig nadeel aan de constructie is dat met het openen van de kast de garantie in gevaar komt. Bij de wat oudere modellen is dit natuurlijk geen groot probleem. Het grote voordeel is natuurlijk dat er geen storende kastjes aan de expansiepoort (aan de linkerkant) hoeven worden aangehangen.
De kaart is in vier verschillende uitvoeringen leverbaar: zonder chips, als 2 MB, 4 MB of als 8 MB kaart. Als optie is een 68881 coprocessor leverbaar. Net als zijn kleinere neefjes is de EXP8000+ autofigurerend, dus regelt zijn eigen installatie. Voor zover bekend is er in Nederland nog geen leverancier van dit produkt.
Inl: (VS) 1-303-8254144

Muziek in Frankfurt

Commodore heeft bekend gemaakt, dat er op muziekgebied, een aantal interessante ontwikkelingen te verwachten zijn. Dat gebeurt tijdens de Frankfurt International Music Fair van 21 tot en met 26 maart in de Frankfurter Messe.
De nieuwtjes zijn te verwachten op zowel op hard- als software te verwachten.

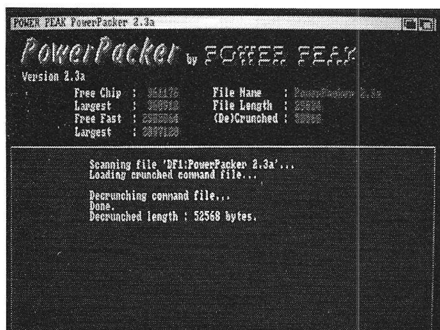
Powerpacker:

Meer files op een disk

Iedere Amiga-gebruiker kent het wel: nog even een laatste commando of programma op die schijf kopiëren en... diskette vol. Het lukt net niet om het over te zetten. Maar dat commando moet er om de een of andere reden toch op. Dan maar een ander deleten en nog maar eens proberen. Een vaak voorkomend probleem: de diskette blijkt te klein te zijn.

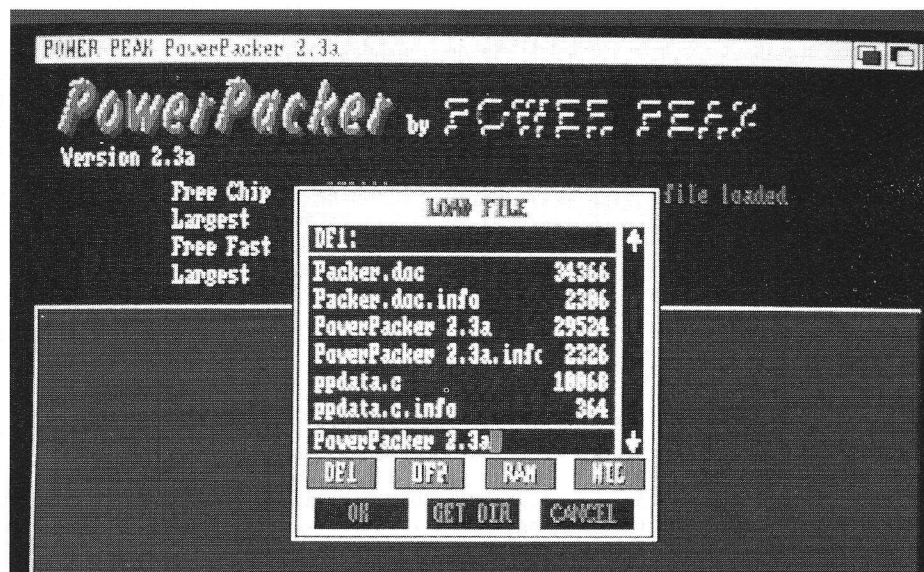
Dat is een probleem waar in de Public-Domain series een oplossing voor is: de cruncher, een programma dat files dan wel programma's crunched, oftewel verkleint. Een meegegeven code zorgt er bij het opstarten voor, dat het programma in het geheugen weer de originele grootte krijgt en normaal werkt. Door nu zoveel mogelijk programma's te crunchen of te 'packen' kunnen er meer programma's op een diskette.

In dit nummer nemen we de belangrijkste en tegelijk nieuwe functies van het Public-Domain programma 'PowerPacker' versie 2.3a onder de loep. PowerPacker werd een aantal jaren door Rob Fonteine's uitgebracht op zijn ANC-diskettes (nu UGA). Inmiddels vinden we het programma in zijn huidige versie terug in de PD-serie van Fred Fish.



Simpel

Crunchen met PowerPacker gaat eigenlijk heel simpel. In het Project-menu alleen maar 'load' klikken en de juiste filenaam in het menu zoeken en intikken, OK en de boel gaat lopen. Rechtsbovenin



beeld staat de naam van de file, de originele lengte en de gecrunchede lengte. In het grote scherm valt precies in procenten af te lezen, hoever het crunchen al gevorderd is.

PowerPacker slaat de verkleinde file op in een buffer. Dat houdt in, dat die na afloop op een diskette moet worden opgeslagen. Dat gaat, hoe kan het ook anders, met de save-opdracht ook uit het project-menu. Daarin staat ook het commando 'Sleep', waardoor PowerPacker op 'stand-by' kan worden gezet. Het programma gaat dan op de menubalk boven in beeld staan. Door er met de muis op te klikken, wordt PowerPacker weer wakker.

De meest nuttige zaken staan eigenlijk in het Prefs-menu. Blijkens de eerste twee opties is er verschil tussen een command- en een data-file. Dat klopt ook wel. Command-files dienen de al eerder genoemde code mee te krijgen waarmee ze automatisch kunnen opstarten en decrunchen. Data-files (bijvoorbeeld tekstbestanden) hebben een dergelijke code niet nodig, vandaar het verschil.

Het decrunchen van een programma in het geheugen kan op verschillende manieren op het scherm worden gevolgd. Welke manier hangt af van de optie 'Decrunch color'. Simpel gezegd kan gekozen worden voor het scherm of de muis. Voor welke manier je kiest, is een kwestie van smaak. Als gekozen wordt voor het scherm, dan verkleurt het beeld tijdens het 'ontpakken'. Bij de andere optie gebeurt datzelfde bij de muis. Amiga-gebruikers van bijvoorbeeld dMouse (laat de muis na een aantal seconden verdwijnen als deze niet wordt aangeraakt) moeten wel in de gaten houden dat het knipperen van de muis tijden het decrunchen niet te zien is.

PPMore

Nieuw in de versie 2.3 is de optie 'Data suffix'. Dit is alleen van belang bij het crunchen van data-files. Wanneer dit punt gekozen wordt, hangt PowerPacker aan iedere file de extensie '.pp'. Op die manier kun je steeds zien welke files gepacked zijn en welke niet. Die extensie is ook erg handig met een nieuw hulpprogramma met de naam PPMore. 'More' of 'less' kennen de meesten wel als tekstviewers. Nu, PPMore is net zoiets, alleen dan voor de teksten die met PowerPacker werden gecrunched.

Encrypt data slaat ook tekst-files. Daarmee kunnen die files van een password worden voorzien tegen onbevoegd inzien.

In het volgende menu 'Recrunch' staan de mogelijkheden voor het ontpakken en een lijst met alle crunchers die PowerPacker ondersteunt. Het menu 'Hunklab' is voor gevorderen, waarbij kan worden bepaald of een programma zich moet decrunchen in het Fast- of Chip-geheugen. En ten slotte 'Script'. Omdat crunchen nogal lang kan duren, kan een draaiboek worden aangelegd. Vooraf moet je dan ingeven welke files op welke drives gecrunched moeten worden en waar die moeten worden bewaard. Zo kan de Amiga bijvoorbeeld 's avonds een hele diskette crunchen, terwijl je zelf met wat anders bezig bent.

Met de nieuwe versie worden ook drie op zichzelf staande programma's meegeleverd, waarvan PPMore er een is. De andere twee zijn 'Crunch' en 'Decrunch'. In feite zijn dit twee kleine PowerPackers. Zoals de namen al aangeven crunched de een en decrunched de ander. Door gewoon het commando op de CLI in te geven, verschijnen vervolgens de opties voor het gebruik in beeld. Deze programma's nemen veel minder ruimte in beslag en kunnen in feite hetzelfde als het complete programma.

W.S.

De hier beschreven versie van PowerPacker staat op Fish 253. De eerdere versies zijn vergelijkbaar, aangezien er per up-date maar weinig veranderingen werden doorgevoerd. Maar een aantal functies van 2.3a zullen op oudere versies niet voorkomen. De afzonderlijke commando's Crunch, Decrunch en PPMore zijn alleen te vinden bij PowerPacker versie 2.3a.

Public-Domain-software

Wat is het?

Goede software voor de Amiga hoeft niet duur te zijn. En dan hebben we het niet direct over de pakketten die door gerenommeerde fabrikanten worden geleverd, maar over programma's die voor een prikkie te koop zijn of zelfs zomaar te krijgen: Public-Domain-software (PD). Vrij vertaald betekent het: 'software die beschikbaar is voor iedereen'.

Het verschil met professionele software is niet alleen de prijs, maar ook het recht van de auteur. De maker van een PD-programma blijft eigenaar, die van een prof-programma verkoopt het meestal aan een software-fabrikant; hij heeft er zo goed als niets meer over te zeggen. Een verder verschil is de handleiding die bij PD niet in boekvorm wordt bijgeleverd, maar meestal als een tekst-file op de diskette staat.
Bij PD kunnen we onderscheid maken tussen 'Freeware', 'Public-Domain' en 'Shareware'. Bij Freeware heeft de auteur alle rechten aan de kant gezet. Iedereen mag het programma gebruiken, het zelfs een andere naam geven, of veranderen. Bij Public-Domain behoudt de auteur wel de rechten over het programma, waardoor het weliswaar gekopieerd mag worden, maar niet ingezet voor commerciële doeleinden. Meestal vragen de auteurs wel een bijdrage, maar aan die vraag hoef je niet persé te voldoen, hoewel zo'n bijdrage wel zo eerlijk is. En ten slotte moet de bezitter van Shareware, wanneer hij ook daadwerkelijk gebruik maakt van het programma, een vast bedrag overmaken naar de programmeur. Wanneer dat niet gebeurt, maakt hij zich schuldig aan diefstal en overtreedt hij het auteursrecht.
In de praktijk worden deze drie groepen voor het gemak in één adem genoemd als 'Public-Domain-software', wat niet inhoudt dat daarmee de verschillen zijn opgeheven.

Verkrijgbaarheid

Hoe kom je nu aan deze veel goedkopere en vaak nuttige software? Daar zijn een aantal verschillende manieren voor. Allereerst zijn er al hele series in omloop bij veel Amiga-gebruikers. Fred Fish PD is daarvan de grootste en bekendste. Verder zijn er firma's die PD-software verkopen, maar ook via een gebruikersgroep is er wel aan te komen. Ten slotte bestaan er in Nederland Bulletin Boards waar je die software met een modem vanaf kunt halen.
Voor beginners is het vaak moeilijk om via kennissen aan PD-software te komen. Het tweede is voor sommigen te duur, want op zo'n PD-diskette staan wellicht ook programma's waar je niets aan hebt. De laatste manier is alleen dan goedkoop als je al een modem hebt. Op die manier krijg je precies wat je hebben wilt.
Blijft over de mogelijkheid een Amiga-gebruikersgroep in je eigen omgeving te zoeken. Bijna al die groepen hebben één of meerdere PD-series, waarvan de programma's tegen geringe kosten te koop zijn. Het geld daarvan vloeit dan terug in de kas van de vereniging. Deze manier is vooral voor beginners meestal de gemakkelijkste.
Tot zover een overzicht van de soorten PD en de kanalen waarlangs deze software zijn weg vindt. In de volgende Commodore Info hopen we een lijst te kunnen publiceren van ons bekende gebruikersgroepen in Nederland. Ben je zelf lid van een groep, of weet je waar er één zit, schijf dan even een briefkaartje aan de redactie.

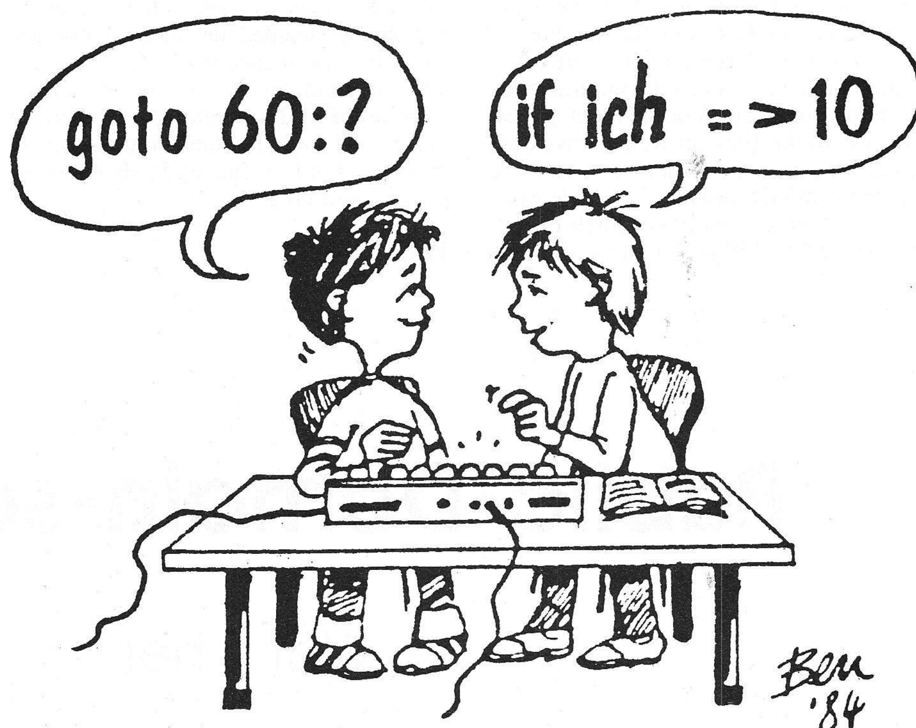
Tip: *Wanneer je veel PD-programma's wilt bewaren, denk dan eens na over de aanschaf van een 5,25 inch floppy drive. Dit formaat diskettes is veel goedkoper dan het gebruikelijke 3,5 inch formaat voor de Amiga en werkt net zo goed.*

On-beveiligen van Basic

Daar zit je dan met je voor *f* 15,- gekochte programma (spel). Er zijn 30 levels mogelijk, maar middenin level-17 (meestal in dezelfde spelsituatie) gaat het fout (altijd). FLITS, wit scherm, dat is het einde. Een of andere leukerd heeft alle kleuren op wit gezet, waardoor iedere melding onzichtbaar wordt en blijft.

Zo begon het artikel van Commodore nummer 1 van dit jaar. Alleen ging er met de samenstelling het een en ander fout. In dit nummer willen we dit graag rechtzetten.
We starten op pagina 65 ongeveer halverwege de tweede kolom, waar staat: if POS 143 then POS = 1. Dit moet zijn: if POS 143 then POS = 1.
Het tweede gedeelte dat voor verbetering vatbaar is, begint op pagina 66 onderaan de eerste kolom met het woordje 'Ik'. Vanaf daar moet de volgende tekst komen te staan:
Ik heb uiteindelijk gekozen voor een recordlengte=256. Van de INPUT-file worden net zo lang records van 256 bytes gelezen totdat er minder dan 256 bytes over zijn. Als nog niet alle bytes van de INPUT-files behandeld zijn, wordt de OUTPUT-file gesloten en weer geopend, maar nu als SEQUENTIAL-file en voor APPEND. Daarna wordt er nog 1 record van 256 bytes gelezen. Hiervan behoren nog maximaal 255 bytes bij de INPUT-file. Deze bytes worden één voor één behandeld en weggeschreven naar de OUTPUT-file.
Bij de definitie van de verschillende tellers dient rekening gehouden te worden met de maximale waarde die ze kunnen krijgen. Short-integers (%) werken sneller dan long-integers (&). De maximale waarde van een short-integer is 32.767 en de maximale waarde van een long-integer is 2.147.483.647. Tellers die de grootte van een file (in bytes) bevatten, moeten long-integers zijn (een file kan meer dan 32.767 bytes bevatten). Tellers die de grootte van een file (in records van 256 bytes) bevatten, kunnen short-integers zijn. De maximale file-grootte is dan 32.767 * 256 = 8.388.352 (ruim 9 diskettes vol) en dat is meer dan voldoende.

De huidige gecompileerde versie doet over 13.000 bytes zo'n 2 à 3 minuten en dat is circa 50 maal sneller dan de eerste ongecompileerde versie.



Public Domain

Van de listings en de daarbij behorende conversie-tabel, is een public-domain-schijf gemaakt, die voor *f* 12,50 (incl. BTW en verzendkosten) te bestellen is bij Infolist, giro 3157656 t.n.v. Infolist Amsterdam, onder vermelding van Unprotect disk. Voor België: stort Bfr. 300 op BBL nr. 310050602562 t.n.v. SAC.
Deze schijf bevat vier bestanden:

° **Read.Me** bevat belangrijke informatie, waarvan ik vind dat men deze gelezen moet hebben alvorens men met het programma Unp van start gaat. Deze file is gecreëerd met ED.

° **Unp** is het met AC/BASIC gecompileerde AmigaBasic-programma, dat m.b.v. de file Unprot.d143 het UNPROTECTEN kan verzorgen.

° **Unprot.d143** is de CONVERSIE-tabel, die onontbeerlijk is voor het programma Unp.

° **Unp.acb** kan ook vanuit AmigaBasic geRUNd worden. Alvorens het programma te RUNnen, dient u precies één keer (niet vaker) het AmigaBasic-commando (CLEAR ,60000) te hebben gegeven.

De listings publiceren bleek niet zo zinvol, aangezien het programma alleen werkt als de CONVERSIE-tabel beschikbaar is. Deze file is 36.608 bytes groot en moet hexadecimaal worden geprint, waarbij voor de leesbaarheid een blank tussen 2 opeenvolgende bytes moet worden gezet. Dit betekent ca. 3 x 36.608 = 109.824 characters. Bij 100 regels van 100 characters betekent dit toch 11 blz. vol. En dan staat het nog niet in de juiste vorm op diskette.

Het compileren zelf maakte het programma maar 2,5 maal sneller (zie ook Commodore 89/4 pag. 72/73).

R.C.A.B. Kohler
Gerrit van der Veenstraat 37
3762 XH Soest

PD-viruskiller

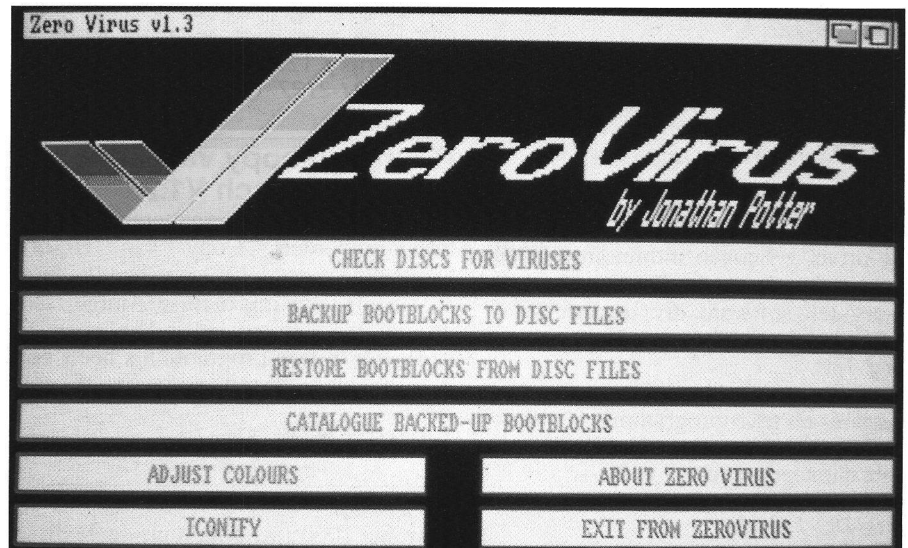
Een goedkoop medicijn

Eigenlijk heeft het altijd al bestaan: kopiëren, om zo goedkoop mogelijk aan goede programma's te komen. Vooral de laatste twee jaar heeft het kopiëren een criminele houding aangenomen. Van programma's zijn tegenwoordig al kopieën op de markt voordat het officieel op de markt is.

Daarnaast is er nog een ontwikkeling ontstaan: de virussen. Eind 1989 (vrijdag 13 oktober) was heel Nederland nog in rep en roer door het zogenoemde 'Data-Crime'-virus, dat na allerlei voorzorgsmaatregelen, zelfs van de overheid, uiteindelijk maar op een paar plaatsen in de wereld opdook. Maar de paniek was niet van de lucht.
Ook de Commodore Amiga ontkomt niet aan het virus-fenomeen. Datzelfde geldt voor het kopiëren van programma's. We hoeven daarbij nog niet eens te denken aan originele programma's. Denk alleen maar eens aan Public-Domain software die op vele gebruikersclub-bijeenkomsten wordt gekopieert. Wanneer eenmaal een virus op een diskette staat en de schrijfbeveiliging staan niet aan, dan vermenigvuldigt het virus zich direct op de andere, nieuwe, diskettes; het is als het ware een epidemie. Virussen verspreiden zich op diskettes wanneer de schrijfbescherming uit staat, dus als je door het gaatje in de diskette kunt kijken. Staat het virus op een diskette waarmee de Amiga wordt opgestart, dan wordt het zeker naar iedere diskette gekopieert die niet beveiligd is.
Met dit in het achterhoofd beginnen we met een korte bespreking van twee programma's die in de Amiga-wereld als 'de beste' te boek staan.

Zerovirus

Zerovirus is een gebruiksvriendelijk programma daar het gebruik maakt van menu's. Na het starten van het programma controleert Zerovirus het geheugen, wat je ook daadwerkelijk ziet gebeuren.



Zerovirus versie 1.3

Daarna komen we in het hoofdmenu terecht. Hier kan gekozen worden tussen het controleren op virussen op een bootsector of virussen die zich aan een file vastklampen, de zogenoemde 'file-virussen'. Andere functies in het hoofdmenu zijn onder andere het maken van een catalogus van alle virussen en het veranderen van de kleuren van het scherm.
Kiezen we voor 'Check disk for viruses' dan komen we in een volgend menu. Links in beeld staan vier disdrives, die afhankelijk van het aangesloten aantal duidelijk of juist onduidelijk staan afgebeeld. Even met de muis er op klikken en de bootsector wordt gecontroleerd. Dat zijn de eerste twee sectoren (0 en 1) van een diskette. In deze 'bootsector' bevinden zich de meeste virussen.Als een virus wordt gevonden vertelt Zerovirus precies welke dat is en staat onderin beeld wat er vervolgens moet gebeuren om de diskette weer virusvrij te krijgen. Bij dit anti-virusprogramma kun je de bootsectoren bekijken. Bij Zerovirus worden de abnormale bytes in rood aangegeven, zodat direct opvalt als er iets niet klopt. Wanneer er met de diskette niets aan de hand is, verschijnt de melding 'Normal DOS-bootblock'.

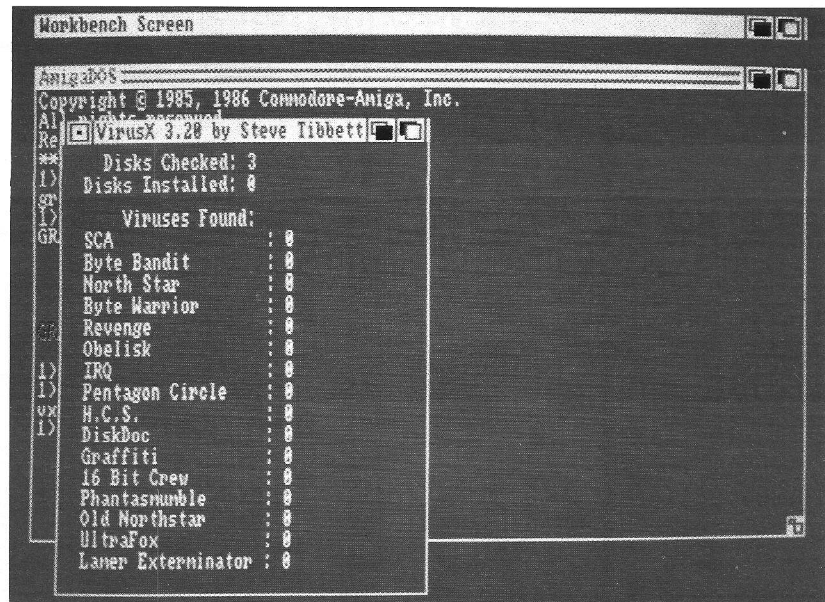
Bootblock keuze

Moet een diskette worden geïnstalleerd, dan kan tussen vier soorten bootsectoren worden gekozen. Die worden ingesteld door met de muis op 'Install Disc' te gaan staan en de rechter muis-toets te drukken. In beeld verschijnt dan een menu met vier mogelijkheden: Standard, NoFastMem, BigScreenTest en AutoAddRAM. De eerste twee zijn meer bekend dan de andere. Standaard is een bootblock dat vergelijkbaar met het Install-commando van de WorkBench-diskette.
NoFastMem schakelt het extra geheugen uit. Sommige programma's kunnen namelijk niet draaien met een geheugenuitbreiding. Iedere Amiga-gebruiker zal wel eens merken dat hij zijn CLI niet helemaal niet naar beneden kan trekken. Dat betekent dat de Amiga op dat moment de NTSC-standaard ondersteunt in plaats van PAL (het hele beeld). Met BigScreenTest kan na het ontdekken van de fout de diskette direct opnieuw geboot worden.
AutoAddRam ten slotte is bedoeld voor geheugenuitbreidingen die niet autoconfigurerend zijn, dat wil zeggen, zich niet automatisch toevoegt aan het standaard geheugen.

Een tweede mogelijkheid van virusbestrijding is een diskette te onderzoeken op file-virussen. Via hetzelfde menu wordt binnen Zerovirus de optie 'File Virus Check' gekozen. Daarna vertelt Zerovirus naar welke file-virussen hij kan zoeken. Vervolgens komt een menu in beeld. Daarin moet worden ingegeven welke files op virussen moeten worden gecheckt. Dit laatste is in vele gevallen een luxe, aangezien de files-virussen lang niet op zo'n grote schaal voorkomen als de bootblock-virussen. Maar toch...
Een aardigheidje van Zerovirus is de 'Iconify'-modus, waardoor het programma verkleint en op de menubalk bovenin de WorkBench komt te staan. Op het moment dat een virus wordt aangetroffen, meldt Zerovirus dat en vraagt of het programma weer actief moet worden om het virus te verwijderen.
Verder kent het programma nog een aantal functies, die we buiten beschouwing willen laten. De hier beschreven functies zijn namelijk voor een effectief bestrijden van virussen ruim voldoende.

VirusX

Het bekendste en waarschijnlijk ook het meest gebruikte programma in deze reeks is VirusX. Het is ook een Public-Domain programma en dus vrij kopieerbaar.
Vroegere virusfinders, -killers of hoe ze ook allemaal worden genoemd, gingen pas aan de slag als daartoe opdracht werd gegeven. VirusX is een anti-virusprogramma van de volgende generatie. Het werkt is beperkte mate automatisch.
Het verdient aanbeveling om iedere programma-diskette die bij een aantal datadiskettes hoort, een tekenprogramma bijvoorbeeld, te voorzien van VirusX.
Het opstarten gaat simpel. Vanuit de CLI simpelweg 'VirusX' ingeven en het programma verschijnt bovenin beeld in de menubalk. VirusX controleert het geheugen, maar komt pas met een melding als er iets aan de hand is. Direct daarna checkt VirusX alle aangesloten drives (bij iedere Amiga tot maximaal vier) op een virus in het bootblock.
VirusX vergelijkt de gegevens die erop staan met een standaard bootsector. Na de controle zijn er drie mogelijkheden.
- er is niets aan de hand;
- het programma vindt een 'non-standard bootblock';
- VirusX vindt een virus.
Bij het eerste hoeft er niets te gebeuren, de diskettes zijn in orde. Bij het tweede en derde geeft VirusX een keuze. Of je trekt je er niets van aan (ignore it) of je vervangt het bootblock door een nieuw (replace it).
Wanneer een virus is gevonden is het logisch de bootsector te vervangen, want wie wil er nu geïnvecteerde diskettes? Bij de tweede mogelijkheid moeten we toch wat voorzichtiger zijn. Sommige programma's hebben namelijk op de eerste twee sectoren een eigen opstartfile staan. Dus als we die bootblocks vervangen, houdt dat in dat het programma niet meer op zal starten. In zo'n geval moet je zeker weten dat dat niet zo is; neem geen risico's. Een bootblock dat eenmaal is overschreven, kan niet meer terug worden gehaald! Dit geldt natuurlijk ook voor andere Viruskillers, waaronder Zerovirus.
Wanneer de muis op de VirusX-balk staat, dat is namelijk het enige dat je ziet (echt een programma op de achtergrond), en je drukt de rechter muistoets, dan verschijnt er een hele lijst met virussen. Het aantal verschilt met de versie die je gebruikt. Daarom is het ook altijd zaak de nieuwste versie hebt, zodat je zeker weet dat tegen alle virussen beschermd bent. In de lijst staat te lezen hoeveel schijven VirusX al gecheckt heeft, hoeveel er al geïnstalleerd zijn en hoeveel virussen het programma al gevonden heeft.
Een andere mogelijkheid is een blik te nemen in de bootblock. Dit gebeurt ook vanuit de lijst, door op '0' te drukken voor df0:, '1' voor df1: enzovoort tot df3:. Bij de wat oudere virussen, zoals SCA, staat daar de tekst 'Your Amiga is alive' te lezen. Bij de latere virussen is het heel wat moeilijker zo niet ondoenlijk virussen op te sporen door alleen maar de bootsector te lezen.
Bij VirusX tot versie 3.2 was het zoeken naar file-virussen nog een probleem, omdat het programma zelf die mogelijkheid niet boodt. Daarvoor werd een ander programma, 'KV' bijgeleverd.
Het opstarten van VirusX kan natuurlijk ook gemakkelijk via de startup-sequence uit de 's'-directory en is zelfs aan te bevelen.



VirusX versie 3.2

Ten slotte

Tot zover een aanzet tot het gebruik van deze twee zeer nuttige hulpprogramma's. Zoals gezegd is het Public-Domain, wat inhoudt dat ze vrij kopieerbaar zijn. Vergeet alleen niet dat bij Zerovirus ook een zogenoemde 'brainfile' hoort, waar alle virusnamen in opgeslagen zitten. Zonder die file herkent Zerovirus geen enkel virus!

W.S.

PAS OP!

Er is een versie 3.3 van VirusX in omloop. Dit is geen viruskiller, maar zelf een virus. De update van 3.2 is 4.0, en dus niet 3.3. Wees gewaarschuwd!
De nieuwste versies van Zerovirus (2.01) en VirusX (4.0) bevinden zich op Fish disk 287. De hier besproken versies zijn respectievelijk gebaseerd op 1.02 en 3.2. Er kunnen dus een aantal verschillen zijn, maar de basis is dezelfde.

Charlie Chip's

SOFT WIR WAR

DOOR:



Live 2000

Een videodigitizer van formaat

Digitizen en vervolgens het bevroren kleurenbeeld bewerken is echt iets voor de artistieke Amiga-fan. Helaas laat de kwaliteit van digitizers nogal eens te wensen over met alle teleurstellende resultaten van dien. Gelukkig is nu de befaamde amerikaanse Live 2000 digitizer geschikt gemaakt voor de PAL- norm. Daarmee komt (semi-)professionele kleurendigitizing binnen het bereik van de videohobbyist.

Videodigitizers dienen om videobeelden te vangen en geschikt te maken voor gebruik in RGB-computer zoals de Commodore Amiga. De gebruiker voert het PAL VHS-signaal in en de digitizer maakt er door de Amiga te bewerken bits en bytes van. Met behulp van paint- of videosoftware kan men het gevangen beeld verder bewerken. Het vangen of bevriezen van het beeld uit de videorecorder of camera vormt de limiterende factor. Freezen kost tijd en de meeste digitizers zullen daarom een stilstaand beeld verlangen en vele tientallen seconden over het digitaliseerproces doen. Hierin blinkt de Live 2000 juist uit. In zwartwit kunnen 15 beelden en in kleur 12 beelden per seconde gedigitaliseerd worden.

Live digitizing

De term "live" slaat bij de videodigitizer Live 2000 op de snelheid van het digitaliseerproces. Echt live kunnen we het nog niet noemen want een tv-signaal biedt immers 25 beelden per seconde aan. Toch komt deze digitizer daar dicht bij in de buurt.
De zwartwitmodus doet een beeld van zestien verschillende grijstinten in 1/15 seconde. Een 32-kleurenbeeld wordt in 1/12 seconde gedigitaliseerd. Daarmee kan de Live 2000 tot de real- time digitizers gerekend worden. En dat voor een prijs van rond de *f* 1.200,-.

De hardware

Het concept van de Live 2000 digitizer is vertrouwd. Een insteekkaart maakt via Direct Memory Access (DMA) rechtstreeks gebruik van het beschikbare Amiga-videogeheugen. Via de vier BNC-bussen kan elke gangbare PAL-videobron worden aangesloten. Eén bus dient als input, een andere als output voor een extra controlemonitor. De beide andere BNC-bussen zijn bedoeld voor aansluiting van een tweede camera. Dat is opmerkelijk omdat de meeste Amiga-videodigitizers slechts met één videobron tegelijk kunnen werken.
Ook is het mogelijk om de BNC-bussen als RGB-ingang voor de signalen Rood, Groen, Blauw en Synchronisatie te gebruiken. Deze modus wordt hardwarematig via draadbruggen ingesteld.
Daar de Live 2000 het Amiga-RAM gebruikt heeft u minimaal 1 MB en liefst 2 MB aan vrij video-RAM nodig. Anders loopt het digitaliseren al snel vast. Een tweede vereiste is een harddisk voor voldoende opslagcapaciteit en het snel kunnen wegschrijven en inladen van de gedigitaliseerde beelden. De digitaliseringssoftware is van huis uit al geschikt voor installatie en het opzetten kost weinig moeite.
De aansluiting van de videobronnen gaf weinig problemen. Alleen bleek de Live 2000 wel gevoelig voor wankele contacten of het wegvallen van het videosignaal. Dan staat er meteen "No videosignal present op de monitor".

De effecten

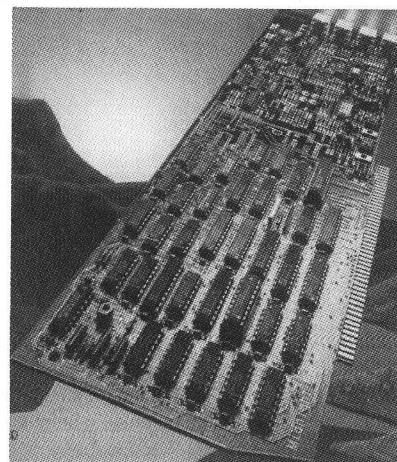
Videodigitizers worden gekocht om de speciale artwork- en video- effecten die daarmee te maken zijn. Bovendien kunnen de eenmaal opgeslagen beelden in animaties en als programmaschermen gebruikt worden. Tot de mogelijke effecten behoren o.a.:

- het inkleuren van zwartwitbeelden. Door verandering van het kleurenpalet zijn alle 16 grijstinten afzonderlijk in te kleuren;
- ook bij de in 32 kleuren of HAM-modus gedigitaliseerde beelden kan men de kleuren volgens de 4096 beschikbare Amiga- tinten selectief veranderen. In de HAM-modus kan ook met een snelheid van vier beelden per seconde gedigitaliseerd worden. Dan valt de Live 2000 eigenlijk nauwelijks meer real-time te noemen;
- gedigitaliseerde beelden kunnen in het IFF-formaat worden weggeschreven en zijn zo gemakkelijk verder te bewerken met paint-software;
- de HAM-beelden kunnen qua scherpte gevarieerd worden
- diverse vervreemdingen zoals, paint-effect, reverse, pseudosolarisatie en beïnvloeding van de beeldkorrel;
- Live 2000 kan kleine animaties opnemen, mits de Amiga over voldoende vrij video-RAM beschikt.

De rest van de effecten wordt aan de fantasie van de gebruiker overgelaten.

Live 2000 is een redelijk tot goede real-time videodigitizer en zal zijn weg naar de grafisch artiest en videohobbyist wel vinden. De prijs is gunstig. Wie betere kwaliteit wil zal aanzienlijk meer moeten betalen.

U.S.



De perfecte Kickstart

Geruchten over 1.4

De Amiga wordt aangezet. Na een enkele seconde verschijnt een geanimeerde diskette op het scherm van een drive, waarin een diskette wordt gestopt en weer uitgehaald. Een nieuwe bootwriter? Er zit geen diskette in de drive... Nee, dit is het nieuwe opstartbeeld van Kickstart 1.4! De nieuwe versie heeft waarschijnlijk binnenkort zijn intrede, zo vernamen wij uit niet-officiële bronnen. Commodore Nederland was niet voor commentaar bereikbaar.

Amiga 1000 bezitters kennen de ergernis van het laden van de Kickstart en daarna de Workbench. Bij de latere modellen van de Commodore Amiga, de 500 en de 2000, werd dit euvel verholpen en de Kickstart in een ROM gebouwd. Het enige dat daarna nog moest gebeuren, was het inladen van de WorkBench.
Bij de 1000 werd nog gebruik gemaakt van de versies 1.0 en 1.1. De nakomelingen kregen een update naar Kickstart 1.2. Maar die hield het maar een aantal jaren vol, toen versie 1.3 zijn intrede deed.
Volgens latere mededelingen zaten er een aantal bugs in de 1.2 ROM, die men er bij de nieuwe versie uit had gehaald. Dat was dus goed nieuws, een bug-vrije versie. Dat daardoor het aantal foutmeldingen in de vorm van een Guru minder werden, zou mooi zijn geweest, maar dat bleek helaas niet het geval.

Voordeel

Voor de doorsnee gebruikers is er met versie 1.3 maar een voordeel gekomen. Kickstart 1.3 ondersteunt namelijk harddisks die autobootable zijn, zodat de machine direct naar een aangesloten harddisk kijkt en daarvan opstart.
De nieuwe Kickstart ging vergezeld van een bijpassende WorkBench met dezelfde soort utilities als bij WorkBench 1.2, maar veel sterk verbeterde 'c'-commando's. Kopers van een Amiga met Kickstart 1.3 bleken helemaal niet zo in hun sas met de bug-vrije ROM. Zij die nog 1.2 gebruikten, konden zonder problemen WorkBench 1.3 draaien en zo ook voordeel trekken van de nieuwe versie. Weliswaar konden 1.3-gebruikers harddisk automatisch laten opstarten, maar sommige leveranciers kregen dit ook met de oudere Kickstart 1.2 voor elkaar.
Het werd nog zotter toen bleek dat een aantal programma's niet meer draaide onder 1.3 die dat onder 1.2 nog wel hadden gedaan. Terwille van hen die een nieuwe Amiga hadden gekocht met Kickstart 1.3 verzonnen een aantal fabrikanten de zogenoemde omschakelaars, waarin een 1.2- en een 1.3-ROM een plaats kregen, zodat tussen beide geschakeld kon worden. Bezitters van een dergelijk apparaatje hebben in de praktijk hun Amiga meer in de 1.2-stand staan dan in de 1.3, notabene de originele ROM!

Kickstart 1.4

Maar nu komt er voor iedereen een sterk verbeterde Kickstart, die een einde moet maken aan alle problemen rond deze systeem-ROM: Kickstart 1.4. Volgens de berichten zal er in maart tijdens de Cebit in Hannover weer een tipje van de sluier worden opgelicht. Maar nu eerst wat er al bekend is van Kickstart 1.4, althans volgens de geruchten.
De meest in het oog springende verandering is het opstartscherm waarop nu een drive te zien is, waarin een diskette gestopt wordt en er vervolgens weer uit komt. Na het opstarten van de, eveneens nieuwe, WorkBench 1.4-diskette en het laden van de daadwerkelijke WorkBench valt op dat het niet langer als beeldscherm, maar als een window wordt voorgesteld. De WorkBench kan dus net als andere vensters worden vergroot ofwel verkleind worden.

ZZZ

Na een dubbele klik op een disk-ikoon wordt het disk-window geopend. De muis, die bij de huidige versies verandert in een slapende wolk (ZZZ), blijft gewoon een pijl waarmee alweer een volgende diskette kan worden geopend. Ook lades die in deze vensters zichtbaar worden kunnen direct met pijlaanwijzer worden geopend. Onder Kickstart 1.4 hoeft wachten dus niet meer. En wat voor het openen van de windows geldt, geldt natuurlijk ook voor het sluiten. Programma's en files die officieel geen ikoon hebben, krijgen er een toegewezen door de nieuwe Kickstart.
Het scherm van de WorkBench heeft boven in de menu-regel een klok en vermeldt daar ook de stand van het geheugen. Verder verdwijnt bij 1.4 in het diskette-window de balk, die aangeeft hoeveel ruimte er nog op de schijf aanwezig is. In plaats daarvan staat er naast de naam bovenin het venster de ruimte in procenten en kilobyte.
Het is de bedoeling dat de gevreesde Guru-meldingen bij Kickstart 1.4 zijn verdwenen. Dat houdt niet in dat er geen foutmeldingen zullen voorkomen. Die worden bij 1.4 aangegeven in de bekende requesters.

Incompatible

De compatibiliteit ten opzichte van 1.2 en 1.3 schijnt nog tegen te vallen. In Duitsland is de nieuwe versie getest en volgens de technici is de incompatibiliteit met versie 1.4 alleen maar vergroot.
Wanneer de chip naar Nederland komt is niet met zekerheid te zeggen. Wel staat vast dat hij komt en net zo simpel te vervangen is als de huidige ROM's.
In hoeverre de geruchten met de waarheid overeenkomen, hopen we binnenkort te melden.

W.S.

Pre-view

Als je al vast een voorproefje wilt nemen van de nieuwe Kickstart moet je de 'Jazzbench' eens proberen van Fish disk 228. Dat is een nieuwe, experimentele, WorkBench die veel doet denken aan de in dit artikel beschreven nieuwe 1.4.

Hama S-590

Meer dan alleen een Amiga genlock

Genlocks zijn een onmisbaar instrument voor de videomaker die het Amiga videosignaal met een camera- of videorecordersignaal wil combineren. Het genlock-interface zorgt ervoor dat alle beeldinformatie uit de beide inputs bij de output op de juiste tijd en plaats belandt. Tijdens de combinatie van het computer en camera- of recordersignaal zijn nog tal van bewerkingen mogelijk. Bijvoorbeeld correctie van helderheid, kleur of contrast en fade- in of fade-out.. Daar blijft het vaak niet bij want de fabrikant voorziet zijn genlock vaak van nog iets extra's. Bijvoorbeeld een geïntegreerde RGB-splitter of ingebouwde videoprocessor. Daarmee wordt de Commodore Amiga plus videosoftware al gauw een kleine montagestudio.Genlocks voor Amiga zijn er te kust en te keur. Er is keuze uit diverse prijsklassen, losse of gecombineerde genlocks en verschillende kwaliteiten. De hier besproken Hama S-590 Genlock is een apparaat dat duidelijk boven de middelmaat uitsteekt. S-VHS- en Hi-8-kwaliteit, combinatie met een RGB-splitter, videosignaalprocessor en decoder voor kopieerbescherming maken deze genlock een aantrekkelijke middenklasser van rond de *f* 1.400,-



Beeldkwaliteit

De **Hama S-590** wil direct hoge ogen gooien en mikt met compatibiliteit voor Super-VHS en Hi-8 op de maximum beeldkwaliteit die in het semi-professionele kanaal haalbaar is. Natuurlijk moet u daar geen wonderen van verwachten. De bandbreedte van het RGB-kleurensignaal op de Amiga is ruim 6 maal groter als bij S-VHS of Hi-8. Er zal dus altijd een achteruitgang zichtbaar blijven. Niettemin doet het gecombineerde genlock outputsignaal niet onder voor de consumentenkwaliteit van S-VHS of Hi-8 video. Gewoon goed bruikbaar in de gevorderde hobby- of semiprofessionele sector.

Behalve de pure genlock-activiteiten doet de Hama S-590 nog veel meer. In de eerste plaats is het interface een prima signaalconverter. Behalve RGB naar Y/C kan de electronica ook Y/C-signalen naar een RGB-SCART monitor omzetten. De beeldkwaliteit is dan net iets minder dan bij puur S-VHS of Hi-8 maar nog wel aanmerkelijk beter dan bij composiet video.

De RGB-splitterfunctie is makkelijk bij het digitaliseren van kleurenbeelden met bijvoorbeeld DigiView. Geen onhandig geklungel meer met gecombineerde opnamen door drie verschillende kleurenfilters. Nu kan alles direct in één keer worden opgenomen.

Als laatste extra een iets minder nette toepassing. De S-590 kan gebruikt worden om de kopieerbescherming van videobanden te verwijderen. Alleen voor eigen gebruik dus en niet voor het grijze of zwarte kopieercircuit.

De aansluiting

Het aansluiten van een genlock is altijd een werkje van goed de gebruiksaanwijzing lezen en de juiste kabeltjes aanschaffen. Doet u dat niet dan gebeurt er niet wat u wilt of vallen de resultaten bar tegen.

Let daarom op de volgende zaken:

- om welke videostandaard gaat het? Gewoon VHS of Y/C S- VHS respectievelijk Hi-8? Elk vergt zo zijn eigen bekabeling;
- welke richting gaat het signaal uit? Sommige goedkopere kabeltjes werken slechts in één richting (d.w.z. weergave of opname) en kunnen dan ook uitsluitend voor dat doel gebruikt worden. Onvolledige bekabeling is een beruchte oorzaak van geen signaal of alleen een zwartwitsignaal;
- het type Amiga. De S-590 werkt met de modellen 500, 1000 en 2000;
- verwijder en plaats alleen kabeltjes met de Amiga op uit!;
- gebruik nooit de RF-uitgang van de video-apparatuur.

De S-590 kiest zelf voor het beste beeldsignaal. Sluit u bijvoorbeeld zowel een gewone Video-in als S-VHS-in tegelijk aan dan kiest het interface in zijn grote wijsheid voor het betere S-VHS beeldsignaal. Verder verdient het natuurlijk altijd aanbeveling om de aansluitingen voor het gebruik nogmaals te controleren.

De genlock zelf

De Hama S-590 zit in een mooi compact rechthoekig plastic & metalen doosje. Gezien de vorm kunt u de genlock alleen naast de Amiga zetten waarbij alle regellaars aan de voorzijde goed bereikbaar zijn. De voeding bestaat uit een losse eenheid. Dat spaart de Amiga-trafo, maar geeft weer een extra kastje, snoertje en stekker op het bureau.
De **FADER**-regelaar verzorgt de aard van superimpose, d.w.z. de manier waarop het ene beeld zichtbaar over het andere wordt heengelegd. In de meest linkse positie werken de faders op de computer graphics. In de meest rechtse positie wordt er van computer graphics naar video en omgekeerd een fade uitgevoerd.
N.B.: Een Fade-in gaat van donker naar licht en een Fade-out van licht naar donker.
De **Mode**-controller heeft alleen een functie in de meest rechtse positie. Namelijk de superimpose van een videobeeld. Zie verder de FADER.
De **INVERTER** kent twee posities. Naar beneden zet het videobeeld in de achtergrondkleur van de computer graphics. In de op-positie verschijnt het videobeeld in de voorgrondkleur. Het behoeft verder geen betoog dat de kleurstelling hier nauw luistert om een zo optimaal mogelijk resultaat te verkrijgen.
De beeldbewerking zelf bestaat uit drie regelaars voor:

- het **kontrast;**
- de **helderheid**;
- de **kleur.**

Verder is er een RGB-schakelaar voor kleur-digitizing aan de monochrome output.

Genlockbeeldkwaliteit en -mogelijkheden

Iedereen die voor het eerst een genlock aansluit wacht een flinke teleurstelling. Het prachtige Amiga RGB-beeld wordt gedegradeerd tot een slap video-aftreksel. Dat is hoogstens ten dele de schuld van de genlock. Het verschil tussen 10 MHz bandbreedte op de RGB- monitor en de 1.5 MHz van de PAL-videostandaard is domweg te groot.
Bij een gescheiden Y/C-signaal (S-VHS en Hi-8) zijn de verliezen qua beeldscherpte duidelijk minder dan bij gewoon VHS of 8mm- video. Bij pure sterk contrasterende kleuren ontstaat echter toch een onrustig beeld. Ook kunnen te fijne beelddetails verloren gaan. Denk daaraan bij de opname en het ontwerpen van de computer graphics. Dat bespaart veel onnodige frustratie.
Tijdens onze test viel op de beeldkwaliteit weinig aan te merken. Midden- tot semiprofessionele klasse. Wie meer wil komt fors duurder uit.
Over de effectmogelijkheden willen we kort zijn. Die bestaan uit aftiteling, combinatie van computertekeningen of animaties met live-video, sleutelgateffect en nachtsimulatie plus wat u zelf nog aan creatieve toepassingen te binnen schiet.

RGB-splitter

Voor het werken met digitizers zoals Digi-View gold en De-Luxe- View is een ingebouwde **RGB-splitter** een heel gemak. In plaats van kleurenbeelden met een zwartwit camera door drie filters heen afzonderlijk op te nemen kan nu alles in één keer met een kleurenacmera. De geïntegreerde RGB-splitter splitst het aangeboden videosignaal in de drie basis-kleuren Rood, Groen en Blauw. Via de RGB-schakelaar stelt u de te digitaliseren kleur in. De digitizer met software doet de rest.
N.B.: In de praktijk zal enig experimenteerwerk nodig zijn om kleurzwemen e.d. te leren elimineren.
De aansluiting is eenvoudig tulp naar tulp vanuit de S-590 naar de digitizer-in-bus. Met behulp van de FADE-knop is het mogelijk om het gedigitaliseerde beeld met het origineel te vergelijken.

Overige toepassingen

De S-590 is geschikt als **S-VHS (Y/C) naar RGB-converter**. Ondanks het feit dat S-VHS alweer twee jaar bestaat ontberen nog tal van tv-sets een geschikte aansluiting. Het S-590 is in staat om het Y/C-signaal zonder al te veel kwaliteitsverlies in een RGB- monitorsignaal om te zetten. U sluit de videobron aan op de S-590 en verbindt de genlock via een SCART-kabeltje met de tv of monitor. Het beeld geeft een duidelijke verbetering ten opzichte van het standaard VHS-videosignaal te zien. Dit alles uiteraard binnen de mogelijkheden van de PAL-norm en iets beneden de echte S-VHS- of Hi-8-kwaliteit.
Het gebruik als decorder voor beschermde videobanden is uiteraard discutabel. Soms kan zoiets handig zijn als u van een zelf gekochte videoband een veiligheidskopie wilt maken.

De S-590 bevestigt de trend naar meer geïntegreerde semi- professionele videosoftware voor de Commodore Amiga. Gevorderde hobbyisten en low-budget semi-professionele gebruikers zullen die *f* 1.400,- er best voor over hebben. Voor dat geld wordt de Amiga een leuke special-effect-en aftitelingsstudio. Info:

Fotec tel.: 05280-62134, Holst Foto tel.: 02290-15241.

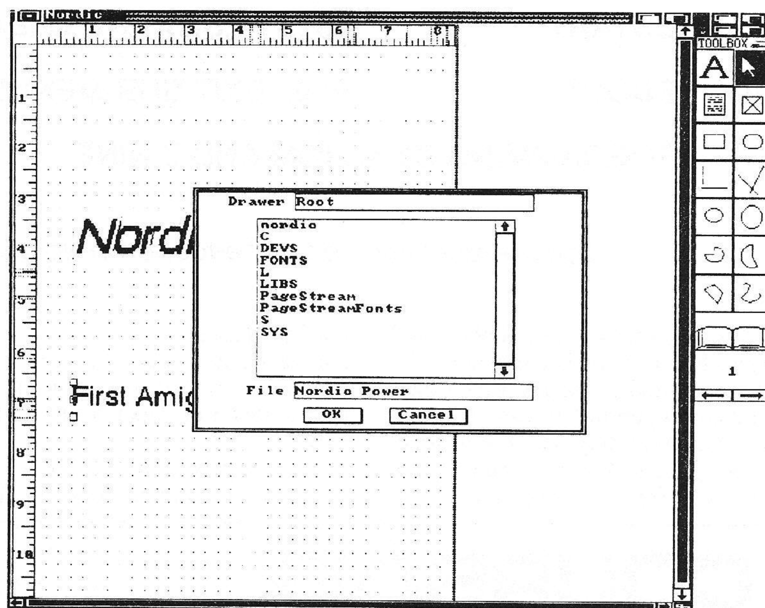
U.S.

Nordic Power

Eerste Amiga Action Cartridge

De Powercartridge voor de Commodore 64 is alom bekend. Minder bekend, maar toch wel veel gebruikt is de Expert-cartridge. Met beide is het mogelijk om programma's die zich in het geheugen van de computer bevinden, naar diskettes te schrijven. Ook spellen die met kopieerprogramma's niet te kraken zijn, worden met deze cartridges zonder moeite gesaved.

Uit eigen land komt nu een dergelijke cartridge voor de Amiga, en wel de Nordic Power, van Data & Electronics uit Venlo. De Nordic Power freezed, maakt screendumps, kent verschillende DOS-commando's, heeft een ingebouwde machinetaal monitor, is voorzien van een doorgevoerde expansiepoort en, last but not least, heeft hij een vertrager die programma's langzamer kan laten lopen.
Veel functies die we graag eens wilden uitproberen.
De Nordic Power is leverbaar in drie verschillende versies: voor de 500, de 1000 en de 2000. Verder is er een uitgebreide 'sc'- en een minder uitgebreide 'lc'-versie. Een positief punt is, dat men de tweede groep niet is vergeten, aangezien er nauwelijks hardwareprodukten voor de Amiga 1000 verschijnen. Wij testten de 500 sc-versie.
De Nordic Power is een klein kastje dat aan de DMA- of expansiepoort van de computer wordt aangesloten. Eventuele andere apparaten, zoals harddisk, kunnen aan de andere kant van de cartridge worden aangekoppeld. De handleiding vermeldt dat niet alle controllers compatible zijn. Verder werkt een Amiga met de nieuwe Fat Agnus chip niet in de PAL-modus.
Achterop de cartridge zitten twee knoppen: een is een draaiknop van de vertrager en de ander is de freeze-knop.



Dit beeld uit PageStream werd met Nordic gefreezed en later met DeLuxePaint bewerkt en uitgeprint

Vertrager

De vertrager kan handig zijn bij spellen die veel te snel gaan om goed bij te kunnen houden. Dat geldt bijvoorbeeld ook voor demo's die beter bekeken moeten worden.
Bij sommige programma's lijkt het of de vertrager niet werkt, maar dan komt hij pas helemaal op het eind in actie. Het ligt er namelijk aan welk programma gedraaid wordt, zo staat ook in de handleiding te lezen. Het ligt er al net aan hoe de routines bij een programma zijn opgebouwd, en hoe de Amiga die afwerkt. Bij de programma's die wij uitprobeerden, reageerde de vertrager steeds laat, maar wel correct.

Freezer

Zodra de freezer-knop wordt ingedrukt, wordt de vertrager uitgeschakeld om het geheugen niet te beschadigen. De freezer-knop doet denken aan die van de Power-cartridge voor de C-64. Een druk op de knop en er verschijnt een grijs menu in de bovenste helft van een, op dat moment, zwart beeld. De Nordic Power werkt constant in het grijze gedeelte van het scherm.
De freezer-knop is weliswaar voor het freezen, maar kan veel meer. Het menu geeft een overzicht van de functie-toetsen:

F1 - Freeze
F2 - Geluid
F3 - Grafisch
F4 - Utilities
F5 - Monitor
F6 - Instellingen
F7 - Herstart
F8 - Clear en reset
F9 - Reboot
F10 - Verlaat submenu

Allereerst nemen we een duik in het freeze-menu. Dat is bedoeld om het geheugen op een of meerdere diskettes te bewaren. Daarbij is het belangrijk of er met een 512 KB of een 1 MB Amiga wordt gewerkt. Zoals bekend kan er op een diskette 880 KB. Dat houdt in dat 1 MB niet op een diskette past.

NORDIC POWER.
FIRST AMIGA ACTION CARTRIDGE V1.8.1SP
COPYRIGHT 1989/90 GAMBIT DEVS., MANUFACTURED BY
DATA & ELECTRONICS, P.O. BOX 3119, NL-5902RC VENLO

F1 ... FREEZER	F2 ... GRAPHICS
F3 ... SOUND	F4 ... DISK-TOOLS
F5 ... MONITOR	F6 ... PREFERENCES
F7 ... RESTART	F8 ... RESET AND CLEAR
F9 ... REBOOT	F10...EXIT SUB-MENUES

PROGRAMMING BY M.ROM AND Z.NINE

Een overzicht van de functie-toetsen in Nordic Power

Als je dus wilt freezen en je hebt 1 MB geheugen, heb je voor het freezen altijd twee diskettes nodig. Data & Electronics beveelt daarom gebruikers van een interne geheugenuitbreiding aan, die uitschakelbaar te maken. Als je tenminste programma's wilt freezen die op een 512 KB machine passen.
Als de F1-toets gedrukt is, verschijnt er een menu, waarin kan worden gekozen tussen saven en laden van 512 KB of 1 MB.
Het saven gebeurt probleemloos en duurt minder dan 1 minuut voor 512 KB. Voor 1 MB geldt een langere duur. Het laden gaat nog sneller. Wanneer een programma eenmaal weer in het geheugen zit, kan je met F7 weer verder werken in het programma, precies waar je op het moment van saven was gebleven. Bij het weer inladen van het programma heb je perse de cartridge nodig. Meenemen naar elders en daar weer opstarten zonder de Nordic Power gaat helaas niet.
De handleiding spreekt niet over meer dan 1 MB. Vuistregel die hierbij moet worden aangehouden, is: hoe groter het geheugen, hoe meer diskettes er nodig zijn.

Geluid

Sommige programma's, maar vooral ook demo's, maken gebruik van geluid. De Nordic Power maakt het mogelijk die samples in het geheugen op te zoeken, naar een diskette te saven, en weer in een muziekprogramma op te vragen. Dat wegschrijven gebeurt in het welbekende IFF-formaat.
Nadat de F3-functietoets is gedrukt, verschijnt er bovenin beeld een lijn van links naar rechts, met daartussen een groot aantal pieken. Dat is de informatie over het geluid die de computer in zijn geheugen heeft staan. Daar kunnen ook bitplanes van tekeningen tussen zitten. Om te con-

troleren waar de geluiden zich bevinden moet de P(lay)-toets worden gedrukt. De computer speelt alle geluiden achter elkaar af. Nu is het zaak even globaal te onthouden waar de gewenste geluiden zicht bevinden. Daarna moet het begin-en eindpunt van de sample worden aangegeven en eventueel de snelheid aangepast, de S(ave)-toets gedrukt en de sample staat op de diskette voor eventuele verdere verwerking.
Bij een nieuwe demo van NewTek (van o.a. DigiPaint III) die we probeerden, zaten diverse geluiden, die zonder problemen door de Nordic werden gevonden en naar een diskette geschreven. Opvragem in Eagis Sonix lukte ook.

Screendumps

Een heel andere, maar vaak gebruikte mogelijkheid is het wegschrijven of uitprinten van schermbeelden. De Nordic kan dat ook. Alllereerst moet een geschikt schermbeeld worden gezocht. Nadat dan de F3-toets is gedrukt, verschijnt er informatie over het beeldscherm. Het beeld ziet er daarna enigszins verwart uit. We volgden bij deze functie de gebruiksaanwijzing op de voet. Maar het viel lang niet mee.
In principe werkt deze functie hetzelfde als die van het geluid, met het verschil, dat de Nordic niet alle plaatjes tegelijk kan laten zien, wat bij het geluid wèl in één keer mogelijk was. Bij deze functie houdt dat in, dat er in het geheugen gezocht moet worden. Daarvoor wordt gebruikt gemaakt van de toetsen 1 t/m 8, waaronder het plaatje kan zitten.
Omdat het menu van Nordic Power zelf ook een plaatje is, kan dat eventueel storen op het scherm. Om dat te kunnen ondervangen, kunnen de toetsen 1 t/m 7 van het nummerieke keypad worden gebruikt. Verder kunnen de verschillende resoluties worden ingeschakeld: lo-res, hi-res, hold & modify, interlace en double playfield.
Bij een aantal programma's probeerden we deze functie uit, wat in sommige gevallen in een hopeloos drukken van toetsen resulteerde. Daarbij ontdekten we dat de NTSC-modus niet direct door Nordic wordt ondersteund. Programma's die in deze modus werkten (in plaats van PAL) gebruiken een gedeelte onderaan het scherm niet. De Nordic gebruikt wel het hele scherm, waardoor het freezen van beelden wel lukte, maar waar onderaan het scherm een wir-war van kleuren en lijnen verscheen, afhankelijk van het plaatje. Nordic laat wel toe dit te veranderen, door die plek zwart te maken. Tot zover geen problemen. Bij deze functie dook wel een paar maal een guru op.
De hardcopies werkten perfect en werden in grijstinten weergegeven. Zoals altijd was het afhankelijk van de kleuren of de afdrukken duidelijk waren. Rood op een donkere ondergrond bijvoorbeeld, komt op een afdruk met grijstinten niet duidelijk over. De rode letters zijn onleesbaar.

DOS-utilities

Eigenlijk kan het bovenstaande niet worden besproken zonder eerst onder F4 te hebben gekeken. Voor het saven van geluid en plaatjes moet namelijk worden aangegeven waar de plaatjes en het geluid naar toe moeten.
Onder F4 bevinden zich een aantal DOS-commando's: dir, path, makedir, delete (remove), format en backup. Bij format maakt de Nordic Power onderscheid tussen Hard en Soft. Hard Format wordt gebruikt bij diskettes die niet eerder werden geformatteerd of onder een ander operating systeem werden gebruikt. Soft Format overschrijft alleen de directory, en is daarom bedoeld voor reeds geformatteerde diskettes onder AmigaDOS.
De andere functies spreken voor zich, alleen path behoeft nadere uitleg. Hiermee moet namelijk ook worden bepaald waar Nordic de geluiden en de plaatjes saved. Wordt dit niet ingegeven, dan zoekt de cartridge altijd op df0:.

Monitor

De monitor van Nordic Power werkt perfect. Alle geheugenadressen zijn bereikbaar en te veranderen. Zoeken naar bepaalde registers levert geen enkel probleem op. Het enige nadeel is het kleine grijze scherm, waarmee Nordic werkt. Hierdoor zijn per keer maar een aantal adressen zichtbaar. Voor de overzichtelijkheid zouden er dit best wat meer mogen zijn.

Overig

Onder F6 zitten een aantal voorkeurinstellingen, zoals de keyboard-layout. F7 is bedoeld om na een freeze, een programma weer verder te laten lopen. Deze toets probeerden we meerdere malen uit. In een paar gevallen bleef de drive hangen en daarmee het programma. Dat lag waarschijnlijk aan het snel achter elkaar stoppen (freezen) van het programma, terwijl de diskette steeds weer een stukje bijlaadde.
F8 en F9 liggen in elkaars verlengde. De eerste maakt het geheugen leeg en start de Amiga opnieuw op. De tweede doet in feite hetzelfde, maar initialiseert de Nordic Power ook. F10 spreekt voor zich.

Handleiding

De gebruikaanwijzing die wordt meegeleverd, is niet veel groter dan de cartidge zelf. De letter is daardoor ook niet erg groot uitgevoerd, waardoor je wel moeite moet doen om het te lezen. Verder is er op nacontrole van de tekst duidelijk bezuinigd, de handleiding stikt namelijk van de taalfouten.
Ondanks deze negatieve punten staan alle functies ruim beschreven. Bij het grafische gedeelte bijvoorbeeld, wordt eerst uit de doeken gedaan, hoe de opbouw van het beeldscherm er uit ziet en de structuur van de IFF-plaatjes.

Ten slotte

Waarom de Nordic Power de eerste Action-cartridge genoemd wordt, is niet geheel duidelijk. Wel is duidelijk dat in Nederland heel serieus wordt nagedacht over de Amiga met deze eerste Nederlandse cartridge. De Nordic heeft een aantal nuttige functies in zich, die voor een aantal gebruikers heel interessant kunnen zijn. Vooral de sample-functie komt erg goed uit de verf. De screendump-functie kan nog iets vereenvoudigd worden, wil deze zonder problemen ook door beginners worden bediend. Maar heb je ook dit onderdeel eenmaal onder de knie, is bevriezen van ieder willekeurig beeld geen enkel probleem meer.
Het freezen van programma's zal waarschijnlijk maar door weinigen worden gebruikt, omdat functioneren zonder de cartridge niet mogelijk is. Vooral gebruikers van 1 MB ondervinden het bezwaar van twee diskettes, om over meer RAM maar te zwijgen.
Het concept is prima en hopelijk gaat Data & Electronics verder met de ontwikkeling van deze soort hardware, waaraan, zo lijkt het althans, niemand nog gedacht heeft.

W.S.

De Nordic Power is te bestellen bij Catronix postorders voor een prijs van f 299,-.